Fakultät für Informatik und Mathematik
Universität Passau, Germany

Tackling cloud compliance through information flow control

Ralph Herkenhöner
Supervisor: Hermann de Meer

A thesis submitted for
Doctoral Degree
July 2015

1. Reviewer: Prof. Dr. Hermann de Meer
   Professor of Computer Networks and Communications
   University of Passau
   Innstr. 43
   94032 Passau, Germany
   Email: hermann.demeer@uni-passau.de
   Web: http://www.net.fim.uni-passau.de

2. Reviewer: Prof. Dr. Gerrit Hornung, LL.M.
   Professor of Public Law, Information Technology Law and Legal Informatics
   University of Passau
   Innstr. 39
   94032 Passau, Germany
   Email: gerrit.hornung@uni-passau.de
   Web: http://www.jura.uni-passau.de/hornung.html

3. Reviewer: Dr. Andreas U. Mauthe
   Reader in Networked Systems
   School of Computing and Communications
   Lancaster University
   LA1 4WA
   Lancaster, UK
   Email: a.mauthe@lancaster.ac.uk
   Web: http://www.scc.lancs.ac.uk/

Abstract

IT outsourcing to clouds bears new challenges to the technical implementation of legally compliant clouds. On the one hand, outsourcing companies have to comply with legal requirements. On the other hand, cloud providers have to support their customers in achieving compliance with these legal requirements when processing data in the cloud. Consequently, the questions arise when IT outsourcing to clouds is lawful, which legal requirements apply to data processing in clouds, and how cloud providers can support their customers on achieving legal compliance.

In this thesis, answers to these questions are given by performing a legal analysis identifying the legal requirements and a technical analysis identifying how legal requirements can be addressed in the context of cloud computing. Further, an information flow analysis is done, resulting in a system theoretical model that is able to describe information flow control in clouds based on the security classification of virtual resources and hardware resources. In a proof-of-concept implementation which is based on the OpenStack open-source cloud platform, it is shown that information flow control can be implemented as a part of cloud management and that legal compliance can be monitored and reported based on the actual assignment of virtual resources to hardware resources. Thereby, cloud providers are able to provide cloud customers with cloud resources, which are automatically assigned to hardware resources that comply with the legal requirements of the cloud customers. This consequently empowers cloud customers to utilise cloud resources according to their legal requirements and to keep control of managing the legal compliance of their data processing in clouds.

Acknowledgements

First of all, I want to thank my parents for supporting me wholeheartedly the entire time. From the very beginning, their love gave me the strength go through all of this.

My particular thanks go to my reviewers for their helpful feedback and for encouraging me on my interdisciplinary research in the field of law and information science. In particular, I thank my supervisor, Prof. Dr. Herrman de Meer, for his continuous support and advice; I thank Prof. Dr. Gerrit Hornung for the supportive discussions on the legal background and his advice; and I thank Prof. Dr. Andreas U. Mauthe for his engagement and the supportive discussions on the technical background.

Also, I thank Wolfang Zimmermann for plenty of inspiring discussions on data protection law in my early research days and Prof. Dr. Norbert Luttenberger for inspiring me to start my research on legal compliance in information science.

Further, I want to thank all my colleagues for supporting me with discussions, feedback, and their friendship; in particular, I want to thank Prof. Dr. Andreas Berl, Focke Höhne, Florian Niedermeier, Michael Niedermeier, Henrich C. Pöhls, and Dr. Patrick Wüchner.

I also thank all the students supporting my research, namely Markus Barth, Sabine Bauer, Christof Blauberger, Berhard Doll, Marcel Ginzler, Kevin Kelpen, Sabine Mayerhofer, Mathias Riediger, and Kai Samelin. Your efforts gave my ideas the wings to fly.

I thank Dirk Emmerich for his supportive feedback on real cloud infrastructures.

Many thanks go to all my friends, who have been supporting me whenever I had to leave early, cancel appointments, and when I was short on the telephone and absent-minded.

Further, I wish to thank Dr. Manfred Kronawitter for his helpful advices.

A big thank you goes to Dr. C. Joanna Sheldon for her helpful remarks on English grammar and orthography and for reading my thesis as a whole.

Last but not least, I want to thank my Alma Mater, the University of Passau, and particularly the Faculty of Computer Science and Mathematics for making all my research work and this thesis possible.

Contents

Chapter 1

Introduction and methodology

IT outsourcing to clouds brings new challenges to the technical implementation of legally compliant clouds. The questions arise as to when IT outsourcing to clouds is lawful and how cloud providers have to support their customers to achieve legal compliance. In this thesis, answers to these questions are identified by analysing legal requirements and technical constraints which are typically applying in the context of cloud computing. Further, a system theoretical model is defined that is capable to describe information flow control in clouds based on applicable legal requirements and technical constraints. In a proof-of-concept implementation, it is shown that information flow control can be implemented as a part of the virtualisation management in clouds and that legal compliance can be monitored and reported based on the actual assignment of virtual resources to hardware resources.

This chapter introduces the topic investigated in this thesis and provides an overview of methodology and results. The following Section 1.1 motivates the topic of IT outsourcing to clouds. The problem description investigated in this thesis is given in Section 1.2. Then, the goal and objectives of this thesis are described in Section 1.3. An overview of the methodology follows in Section 1.4. In the end, Section 1.5 provides an overview of the structure of this thesis, and Section 1.6 summarises the scientific contributions.

1.1 Prospects of IT outsourcing to clouds

Many organisations seek to outsource their IT infrastructure and IT processes partially or fully. For this purpose, an organisation contracts a service organisation having the desired expertise and offering the requested services. The degree of IT outsourcing can range from outsourcing the operation of single IT systems (like servers) to the operation and maintenance of the IT infrastructure including servers and workstations. IT outsourcing can also cover the execution of IT processes (like accounting) by service organisations. Both operation of IT infrastructure and execution of IT processes by a service organisation are generally denoted outsourced services.

A common reason for outsourcing is the reduction of operating costs. Here, service organisations which specialise in the cost- and/or time-efficient operation of IT infrastructures and execution of IT processes can help to reduce operating costs. For example, a data centre can operate their customers' servers more efficiently since the support infrastructure (including fa-

cility, cooling and power supply) is utilised by multiple customers. An other reason is that the operation of IT infrastructures and execution of IT processes often require expert knowledge which usually is not part of the business operations of outsourcing organisations. In this case, a service organisation having such expert knowledge is capable of supporting outsourcing organisations with professional services they otherwise couldn't handle on their own (or at least not on the same professional level).

Particularly difficult to serve are service demands that are non-continuous and occur in response to specific events or circumstances (e.g., annual accounts and release dates of pre-selling contingents with great demand). In such cases, outsourced services have to be provisioned in a timely manner and in a way proportionate to the demands of the outsourcing organisation. In particular, the IT infrastructure necessary for serving such on-demand services has to scale properly to avoid shortages and interruptions. Here, cloud computing represents a service model that addresses the provisioning of on-demand services in a flexible and scalable manner. In this context, service organisations are cloud providers operating the cloud infrastructures hosting the outsourced services, and the outsourcing organisations are cloud customers. To address flexibility and scalability in cloud computing, available hardware resources are pooled using virtualisation techniques which make it possible to provision proportions of computing resources to multiple cloud customers (i.e., outsourcing organisations). The cloud customers can order cloud services (e.g., virtual servers or applications) on demand and in a self-service manner. Requested cloud services are then provisioned automatically according to the cloud customer's needs.

Cloud computing combines the flexible and scalable usage of computational power (as in grid computing) with automated on-demand self-services making it a good candidate for IT outsourcing. There are three distinguished services models in cloud computing that can be utilised by the cloud customer for IT outsourcing: (1) Infrastructure-as-a-Service (IaaS), i.e., the operation of single IT systems up to complex IT infrastructures in clouds (like Amazon Web Service (AWS) [5] and Microsoft Azure [143]), (2) Platform-as-a-Service (PaaS), i.e., the development and provisioning of cloud services (like Google App Engine[1] and Apprenda[2]), and (3) Software-as-a-Service (SaaS), i.e., cloud hosted applications (like Google Docs[3] and Cisco WebEx[4]). Thus, cloud computing covers a large number of possible outsourcing scenarios.

However, there are many organisations hesitating to outsource their IT infrastructure and IT processes to clouds. Instead, these organisations prefer traditional outsourcing to dedicated data centres and to service organisations not utilising cloud computing. There exist multiple reasons why organisations decide not to outsource to cloud infrastructures. A major reason is that organisations have to comply with legal obligations that apply when processing their data. For instance, European organisations have to comply with European data protection law which explicitly regulates the transfer of data to third countries and particularly stipulates an *adequate level of protection* at the target location (Art. 25 Data Protection Directive). Further, there exist specific requirements on confidentiality (e.g., German professional secrecy, §203

[1] Google App Engine Documentation, on the Internet: https://cloud.google.com/appengine/docs (last visited: 30.06.2015)
[2] Apprenda, on the Internet: http:\www.apprenda.com/ (last visited: 30.06.2015)
[3] Google Docs, on the Internet: http://www.google.com/docs/about/ (last visited: 30.06.2015)
[4] Cisco WebEx, on the Internet: http://www.webex.de/ (last visited: 30.06.2015)

Strafgesetzbuch (StGB)), integrity (e.g., German tax data must not be modified, §146 para. 4 Abgabenordnung (AO)), and availability (e.g., German personal data are subject to availability control, cl. 2 no. 7 of appendix to §9 cl. 1 Bundesdatenschutzgesetz (BDSG)), which have to be ensured during data processing including the outsourcing to clouds. Additionally, there exist regulations which limit or prohibit IT outsourcing. For example, in Germany, accounts have to be kept and stored within Germany, and outsourcing is subject to authorisation (§238 Handesgesetzbuch (HGB)). Outsourced services have to comply with these legal obligations. This is difficult to achieve when outsourcing to clouds.

On the one hand, many cloud providers operate cloud infrastructures in multiple countries (e.g., AWS, Microsoft Azure, Fujitsu Cloud, IBM Cloud), and therefore, cloud services may be hosted in different countries, which can be problematical if requirements on transfer control apply. Particularly for European organisations outsourcing personal data, cloud providers located outside of the European Union/European Economic Area (EU/EEA) are problematical since transfer of personal data to third countries is only admissible if an *adequate level of protection* is ensured. There exist reasonable doubts whether this is generally the case for cloud providers which are established outside of the EU/EEA [29].

On the other hand, outsourcing organisations are generally responsible for the IT outsourcing and often are obliged to inspect the legal compliance of service organisations, for example, in the context of European data protection law (Art. 7 lit. a in association with Art. 6 para. 1 Data Protection Directive) and in the German financial sector (§25a para. 1 cl. 5 Kreditwesengesetz (KWG)). However, the inspection of cloud providers and cloud infrastructures by outsourcing organisations is difficult. If the cloud infrastructure is hosted in multiple data centres which are located in multiple countries then the outsourcing organisation has to inspect each data centre individually which requires considerable efforts. Additionally, expert knowledge on the operation of cloud infrastructures is required at the outsourcing organisation which possibly is not the case and also requires additional efforts. Further, inspections of data centres by every single outsourcing organisation can create considerable overhead and can even cause disturbance to the secure operation of data centres if there is a huge number of outsourcing organisations (which is usually the case for large cloud providers). Another issue is that the inspection has to be done individually for every outsourced service. Due to virtualisation, the inspection of outsourced services requires the identification and inspection of each provisioned virtual resource and each related hardware resource that is utilised. Here, cooperation of the cloud provider is necessary, since the outsourcing organisations have no insight into the virtualisation management and particularly not the utilised hardware resources in clouds.

Consequently, in the case of IT outsourcing to clouds, outsourcing organisations become cloud customers and are in need of legally compliant processing of their data in the cloud. To address this need, cloud providers have to be able to support their customers in their efforts to achieve legal compliance by processing data within clouds according to applicable legal requirements and by enabling the monitoring and reporting in respect to data processing within clouds. The importance of achieving legal compliance in clouds is underlined by the fact that cloud computing has become a critical infrastructure [67].

1.2 Problem statement on legally compliant cloud computing

The case of IT outsourcing to clouds incorporates legal and technical challenges. On the one hand, there are legal requirements that apply to the data processing of cloud customers and to IT outsourcing generally. These legal requirements have to be supported by cloud providers when processing data on behalf of their customers. On the other hand, cloud computing is highly automated using virtualisation techniques and distributed computing. In order that cloud providers can support the legal requirements of their customers and IT outsourcing, these requirements have to be translated into technical requirements, which are then implemented in cloud infrastructures. Consequently, the questions arise, which legal requirements apply when outsourcing IT to clouds and how can they be supported in cloud infrastructures? To answer this question, it is necessary to investigate both applicable legislation and technical capabilities of cloud computing.

A repetitive legal requirement is the regulation of processing abroad, which often is treated differently from processing inland. For example, European data protection law requires an *adequate level of protection* if personal data are processed outside of the EU/EEA (Art. 25 Data Protection Directive), and German tax law restricts the outsourcing of accounts to foreign countries (§238 HGB). Consequently, in the case of IT outsourcing to clouds, this type of requirement – beside others identified in this thesis – has to be supported by the cloud provider. If a cloud customer orders a cloud resource (e.g., a virtual machine) that, for instance, is only to be hosted in Europe then the cloud provider has to ensure that only data centres located in Europe are commissioned to host this cloud resource. Since resource management in clouds is fully automated the question arises, how customers' requirements can be translated to security policies which are enforced automatically by only assigning data centres complying with the customers' requirements. The same is true for legal requirements addressing security constraints in respect to confidentiality, integrity and availability. In this context, multiple key questions arise: How does the cloud provider know which data centre provides the required level of security and how are data centres selected by automated resource management accordingly? Furthermore, how do cloud providers monitor and report on compliment resource assignments if they are legally obligated to do so (e.g., in the context of German data protection law), and what is necessary to ensure that a cloud customer is able to inspect a cloud provider on legally compliant data processing? These questions have to be answered when outsourcing data processing to clouds. This thesis identifies possible answers by investigating legal compliance of IaaS clouds in the context of European and particularly German legislation.

1.3 Goal and objectives of the thesis

The goal of this thesis is to define the legal and technical boundary of IT outsourcing by European and German cloud customers in the case of globally distributed cloud computing with a particular focus on location-determined resource management.

To achieve this goal, there are several objectives that have to be fulfilled. They can be categorised by (1) understanding the mutual dependencies of legislation and technology in the cloud computing context and by (2) the information theoretical implementation of location-

determined data processing in cloud computing environments. They are defined as follows.

Understanding the mutual dependencies of legislation and technology:

- **Objective 1:** Identification of the legal requirements on IT outsourcing using cloud computing (in the specific case of European and German legislation).
- **Objective 2:** Identification of the capabilities of technical implementations of cloud computing (considering current and future requirements).
- **Objective 3:** Understanding the technical implications of the legal requirements on the utilisation, provisioning, and hosting of Infrastructure-as-a-Service (IaaS) for IT outsourcing.
- **Objective 4:** Identification of the legal and technical boundary implicated by established legislation and the cloud computing paradigm.

Information theoretical implementation of location-determined data processing:

- **Objective 5:** Modelling legal and technical requirements with respect to the security goals of confidentiality, integrity, availability, and location determination, including
  - Extension of existing concepts by availability and location determination **(Item 5.1)**; and
  - Introduction of new concepts for location determination **(Item 5.2)**.
- **Objective 6:** Implementation of location-determined data processing within cloud resource management, including the monitoring of compliance with legal and technical requirements by the cloud provider and the reporting on compliance to the cloud customer.
- **Objective 7:** Identification of the feasibility of trustworthy compliance enforcement, monitoring, and reporting within cloud infrastructure (by the cloud provider to the cloud customer), including:
  - Demonstration of the feasibility of location-determined data processing and compliance monitoring and reporting in cloud computing environments **(Item 7.1)**; and
  - Evaluation of the achievable scalability, reliability, trustworthiness, and legal compliance of location-determined data processing **(Item 7.2)**.

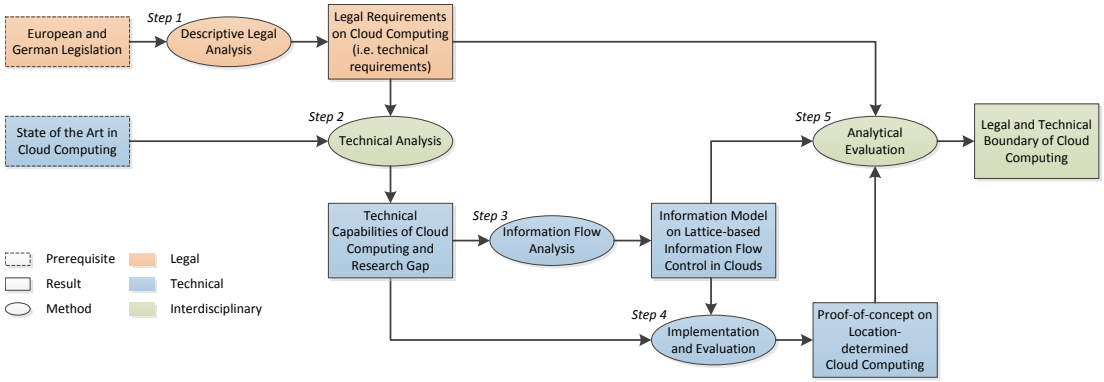
1.4 Methodological approach

The methodological approach chosen in this thesis aims to achieve the goal of defining the legal and technical boundary of IT outsourcing to clouds (cf. Section 1.3). This requires the investigation of legal requirements applying to IT outsourcing to clouds and the technical

capabilities of cloud computing to address applicable legal requirements. For this reason this thesis follows an interdisciplinary approach covering both legal and technical investigations.

Generally, the methodological approach of the thesis consists of five sequential steps which are illustrated in Figure 1.1. In a first step, a legal analysis is performed identifying relevant legal norms and legal requirements for IT outsourcing to clouds. Then, in a second step, the technical capabilities of supporting the identified legal requirements are analysed. The result is a number of technical requirements, which has to be implemented in clouds, and to which extend these requirements are addressed by current research. A finding of the second step is the lack of modelling security and particularly location constraints and their enforcement in cloud resource management (cf. Section 4.4). Also a result of the second step is a general terminology and entity-relationship model describing cloud infrastructures using an ontology's notion. In a third step, based on the methods of information flow control, an information model is defined that classifies information flows in cloud infrastructures by security and location constraints and thereby allows the control of information flows within cloud infrastructures. The forth steps shows the feasibility of the information model by practical evaluation of a proof-of-concept implementation based on the open-source cloud platform OpenStack. In a final step, the legal and technical boundary is concluded in an analytical evaluation comparing the findings of the legal analysis with the observations made on the technical implementation. The legal and technical methods used in these steps are explained in more detail in the following.

Figure 1.1: Methodological approach of the thesis.



The *legal part of the thesis* covers a descriptive legal analysis of legal norms corresponding to IT outsourcing to clouds in European and particularly German legislation. The goal of this analysis is the identification of those requirements, which have an impact on the technical implementation of cloud computing. To make such a statement, it would be theoretically necessary to investigate all corresponding legal norms in the EU/EEA and Germany. This is however not feasible because of the large number of existing legal norms within EU/EEA and Germany. Moreover, a complete analysis is not necessary to identify technical requirements that are necessary for implementing legally compliant clouds, since it is sufficient to understand which general requirements are addressed by law and which options are possible. A good starting points for the legal analysis are legal norms with particular importance for data processing

and IT outsourcing. In this thesis, the legal requirements are exemplified by data protection law and export control on a European and German level and additionally by German sectoral requirements in finance, taxation, medial and healthcare, and public agencies. Thereby, multiple common cases of IT outsourcing are covered, which consequently also apply to cloud computing. To identify the requirements and possible options, a descriptive legal analysis is made. Resulting from the analysis are a number of legally defined technical requirements, which apply to clouds in the context of IT outsourcing.

The *technical part of the thesis* consists of (1) a technical analysis of cloud computing and its support of legal compliance, (2) the definition of an information model addressing the identified research gap, and (3) the implementation and evaluation of the findings in this thesis. The goal of the technical part is the technical implementation of legally compliant clouds. Therefore the technical capabilities of cloud computing are investigated. The result of the analysis of cloud computing is a mathematical model describing entities and relations in cloud infrastructures. It is defined using a notion for ontologies to enable its usage on a formal basis. The entities and relations in the model are derived from the National Institute of Standards and Technology (NIST) cloud reference architecture [134] and by the analysis of five prominent cloud platforms in practice. The model forms the basis for the cloud terminology used in this thesis. Further, it is the basis for the analysis of supported legal compliance in clouds and for the information model formulated in this thesis. The analysis of supported legal compliance in clouds investigates the state of the art in science and practice with respect to the ability to address the technical requirement identified in legal analysis and compare it with the requirements identified in the legal analysis. The result of the analysis is a research gap on dealing systematically with security and location constraints of cloud customers to support legal compliant data processing in clouds. The identified research gap is addressed in this thesis by an information model which is able to describe security and location constraints of cloud customers and can be used in the cloud management process to enforce legally compliant resource allocation within cloud infrastructures. To define the information model, a system theoretical approach on information flow control is used. For this purpose, information flows in clouds are identified based on the cloud resources in cloud infrastructures. This allows the classification of cloud resources by security and location constraints and make it possible to decide and enforce on information flows between cloud resources. The validity of the information model is shown by proving the security properties of the system controlling the information flows and by the description of a theoretical approach to implement information flow control in cloud management. The feasibility of practical application is shown through experimental and analytical evaluation. The experimental evaluation covers a proof-of-concept implementation and a test of its ability to support location constraints in the assignment process of virtual machines. For the proof-of-concept implementation, the open-source cloud platform OpenStack and its extensions for resource pooling and logging were modified. In the analytical evaluation, the ability to comply with legislation and service-level agreements (SLAs), technical feasibility with respect to cloud computing characteristics, and the trustworthiness of implementations are reflected. As a result of the analytical evaluation, the legal and technical limits in legally compliant cloud computing are identified describing a legal and technical boundary of cloud computing which is the ultimate result of this thesis.

1.5 Structure of the thesis

The current chapter motivates the context of the topic of the thesis (cf. Section 1.1) and defines the description of the problem (cf. Section 1.2) as well as the goal and objectives of this thesis (cf. Section 1.3). Further, the underlying methodological approach (cf. Section 1.4) and the structure of this document (cf. Section 1.5) are explained. In addition, the scientific contribution of this thesis is described, highlighting major findings and the delta in research (cf. Section 1.6). The remainder of this document is structured as follows.

Chapter 2 provides background on cloud computing (cf. Section 2.1), IT outsourcing to clouds (cf. Section 2.2), and legal compliance (cf. Section 2.3). Further, research work related to the thesis topic is summarised (cf. Section 2.4) and major aspects of legally compliant cloud computing are concluded (cf. Section 2.5).

The legal analysis is performed in Chapter 3. A general analysis of the lawfullness of cloud computing is provided in Section 3.1. Subsequently, requirements of German data protection law is discussed, as it applies to carrying out data processing (cf. Section 3.2). In Section 3.3, the intermediate result of necessary safeguards at the cloud provider is presented. Then, the sectoral requirements are exemplified on the basis of European and German export control and German legislation on finances, taxation, medical care and healthcare, and public agencies (cf. Section3.4). Section 3.5 identifies special requirements which are of particular importance when outsourcing to clouds. Section 3.6 concludes the identified technical requirements in European and German legislation.

Chapter 4 covers the technical analysis of cloud computing and its ability to support achieving legal compliance. In Section 4.1, cloud terminology and cloud entries and their relations are defined by deriving an IaaS cloud computing ontology. The technical analysis of clouds' ability to support achieving legal compliance is then made with respect to security management (cf. Section 4.2) and compliance management (cf. Section 4.3). The identified research gap and the implications on implementing legally compliant clouds are concluded in Section 4.4).

Chapter 5 deals with the system-theoretical approach to information flow control in clouds. The information flows in clouds are identified in Section 5.1. Existing research on information flow control is investigated in Section 5.2. In Section 5.3, a model for information flow control in clouds is presented. Section 5.4 explains how information flow control can be implemented in clouds, followed by conclusions in Section 5.5.

The experimental and analytical evaluation is done in Chapter 6. The description of the proof-of-concept implementation and the experimental set-up and results are presented in Section 6.1. Section 6.2 covers the evaluation in respect to legal compliance, technical feasibility, and trustworthiness of implementations. Section 6.3 concludes the legal and technical boundary of legally compliant cloud computing.

Chapter 7 concludes the results and findings of this thesis, reflects on application and practical implications, and provides an outlook on future research topics.

The appendix contains (i) details of the proofs made in Chapter 5, (ii) the example policies and the underlying XML-schema used in the experiment, (iii) the logs and screen-shots of the experiment, (iv) the glossary, the acronyms and symbols used in this thesis, (v) lists of figures, tables, listings and definitions, and (vi) the references.

1.6 Scientific contribution

This thesis contributes in multiple ways to the research on legally compliant IT outsourcing to clouds. The most significant contributions are listed in the following:

- The systematic identification of **technical requirements** that are defined **in legal norms** of European and German legislation and which are generally **applicable to IT outsourcing to clouds**, including detailed inspection of necessary safeguards at cloud providers and on location-determined data processing. Particular findings are (1) the identification of the *necessary level of security*, (2) the usage of security policies, (3) the implementation and enforcement of safeguards (i.e., basic security measures, access control, transmission control, and countermeasures and indecent response), and (4) the monitoring, documentation and reporting of compliance (cf. Section 3.6).

- The definition of a **taxonomy and entity-relationship model on cloud infrastructures** in an ontology's notion, which is able to map virtual resources with hardware resources using mathematical functions on sets of the respective resources (cf. Section 4.1).

- The **combination of the two basic models on information flow control** by Bell and La Padula and Denning into a single model (cf. Lemma 5.1) and its adoption into the integrity-based model by Biba (cf. Lemma 5.2).

- The definition of a **general model on latices-based information flow control**, which is provably secure (cf. Theorem 5.5) and supports general security characteristics including confidentiality, integrity, availability and location constraints.

- The definition of a **lattices of location classes** introducing location constraints to the methods information flow control (cf. Section 5.3.2).

- The definition of a **lattices of availability classes** introducing availability to the methods information flow control (cf. Section 5.3.3).

- The application of of **lattice-based information flow control on virtual resources** in clouds (cf. Section 5.3.4.1) and its implementation in the cloud management process (cf. Section 5.3.4.2) using security policies (cf. Section 5.3.4.2). In particular, information flow is control on the basis of the security classification of virtual and hardware resources.

- The definition of a **security-class-based metric for calculating optimal resource allocations** in clouds (cf. Section 5.4.2), which covers multi-dimensional security requirements (i.e., confidentiality, integrity, availability, and location).

- The proposal of an **architecture for compliance monitoring and reporting** in clouds (cf. 5.4.3), which enables automated monitoring of and reporting on the assignment of virtual resources to hardware resources in respect to their security classifications.

- The **implementation of the Location-Determining resource management and logging Architecture (LDA)**, which is a proof-of-concept on location-determined data processing in clouds using the open-source cloud platform OpenStack (cf. Section 6.1).
- A particular finding of the the legal analysis is the **identification of a conflict of German tax law with the technical reality of cloud computing**. German tax law obliges that there is direct access to the IT systems during a tax inspection (recital 165, GoBD). This does not fit with the basic concept of remote enquiry in cloud computing, which is the opposite of direct access (cf. Section 3.4.2).
- The **extension of the existing classification of cloud computing** into a service and deliver model through the new dimension of hardware locality, which distinguishes between national and global clouds specifically (cf. Remark 2.5).
- The analysis and proposal of a **trustworthy classification of hardware resources** in clouds (cf. Section 5.4.1), which provides a basis for lattice-based information flow control on virtual resources in clouds.

Chapter 2

Cloud computing and legal compliance

This chapter provides the necessary background for understanding the investigations on legally compliant cloud computing made in this thesis, In Section 2.1, a basic overview of cloud terminology is given, including concepts of utilisation and alternative approaches to outsourcing. Preparing the legal analysis in Chapter 3, Section 2.2 introduces actors and their interaction in the context of outsourcing to clouds and motivates the relevance of legal compliance for cloud customers and cloud providers. General observations on technical characteristics of clouds and their impact on legal compliance are introduced in Section 2.3. These observations form a supportive basis for the legal analysis in Chapter 3. Section 2.4 summarises the related work identified by this thesis followed by conclusions on legally compliant clouds in Section 2.5.

2.1 The cloud computing paradigm

In this section, the necessary background and terminology of cloud computing for this thesis is introduced. Therefore, a definition of cloud computing based on existing definitions is formulated. Further, concepts of utilisation and their impact on legal compliance are discussed in the context of IT outsourcing. Additionally, alternative approaches for IT outsourcing are investigated, and in particular, similarities and differences with respect to legal compliance are discussed.

2.1.1 Towards a comprehensive definition of cloud computing

The current understanding of cloud computing has emerged from two broadly accepted definitions of cloud computing, manifested in most of the existing cloud computing implementations. The first is defined by Vaquero et al. and provides a consolidated definition of existing concepts and implementations of cloud computing in 2008 [211]. The second is defined by the NIST and represents the core definition of the NIST's cloud computing standards [141], and for example, is also adopted by the European Network and Information Security Agency (ENISA) [70, p. 9] and the Cloud Security Alliance (CSA) [46, pp. 11 et seqq.]. Both definitions are compatible with each other and can be merged to a single, more comprehensive definition as follows.

Definition 2.1 (Cloud computing) *Cloud computing is a concept for hosting, provisioning, and utilising a shared pool of virtualised computing resources in a scalable, ubiquitous, on-demand, and easily usable manner. The computing resources are accessible via network, paid by usage, automatically and rapidly provisioned and released with minimal effort and interaction, and have guaranteed characteristics defined in customised SLAs.*

Remark 2.1 (Hosting, provisioning, and utilisation) *While the original definitions focus on the utilisation of computing resources and their characteristics [211][141], cloud computing also covers the hosting and provisioning of computing resources. Hosting and provisioning are implicitly covered by the mentioned definition, since defining characteristics of computing resources also includes implications on their hosting and provisioning. Without loss of generality, the explicit extension of the existing definitions makes it possible to address additionally the technical aspects of cloud hosting and provisioning which are relevant for the investigation of legally compliant cloud computing (cf. Section 3.6).*

Of particular relevance for investigating the question of how to implement legal compliance in clouds are the technical characteristics of cloud infrastructures and how implementing legal compliance is affected by these characteristics. In particular the question arises, how much cloud computing will be left after implementing legal compliance. This requires an investigations of existing cloud characteristics and the possible impact of implementing legal requirements. As an extension to the Definition 2.1, the following characteristics are considered essential to cloud computing. They are derived from the definitions given by NIST and Vaquero et al.. Based on these characteristics, the analytical evaluation in Section 6.2 investigates the impact of implementing legal compliance in clouds.

2.1.1.1 Resource pooling [141] | virtualisation [211]

The computing resources at the hosting site are virtualised and pooled to provision portions of computing resources to multiple cloud customers in consideration of multitenency. In general, the hosting process is technically transparent (i.e., unrecognisable) for the cloud customer. For example, the location of the provided resources is unknown to the cloud customer. In any case, virtualisation can introduce notable side effects like network delay (e.g., caused by the overhead of virtual networks). The computing resources are specifically defined within the cloud computing ontology in Section 4.1. Resource pooling and virtualisation are of particular relevance for cloud computing since they are the technological enabler of cloud computing and imply the abstraction of cloud resources from hardware infrastructure. The impact on legal compliance is investigated more spefically in Section 2.3.3.

2.1.1.2 Rapid elasticity [141] | scalability [211]

Computing resources are provisioned and released elastically and scale according to the cloud customers' demands. Albeit the physical computing resources are limited, the provided computing resources can appear to be unlimited and flexibly configurable to virtually any amount or size. With respect to legal compliance, the question arises whether enforcing legal constraints

results in limiting the elasticity and scalability, because available resources cannot be utilised due to legal constraints.

2.1.1.3 On-demand self-service [141] | pay-per-use utility model [211]

Computing resources are automatically provisioned as needed if and only if the cloud customer requests them. The payment is based on the amount of usage (e.g., by capacity or time). When implementing legal compliance, this characteristic may be influenced by additional efforts to ensure legal constraints. Two questions arise: whether accounting of additional efforts is possible and whether provisioning resources automatically is still feasible.

2.1.1.4 Measured service [141]

Resource management automatically measures and controls the operation of cloud resources (e.g., optimising hardware utilisation, monitoring health, and measuring usage). It is possible to report the use of computing resources to both the cloud provider and the cloud customer. This characteristic is of particular interest when it comes to gauging the legal compliance of services. The feasibility and trustworthiness of such measurements are paramount for reliable results.

2.1.1.5 Broad network access [141]

The computing resources are accessed through standard mechanisms of network communication (e.g., Internet) via heterogeneous end systems (i.e., client platforms like mobile phones and workstations). This characteristic is highly important to ensure the availability and accessibility of cloud resources by cloud customers. This is because cloud customers usually utilise cloud resources from remote locations requiring network access. Two questions arise: whether there exist legal constraints that hinder access from remote location and how such legal constraints should be dealt with.

Remark 2.2 (Technical implications of cloud characteristics) *All considered cloud characteristics have technical implications for the implementation of legal compliance in the context of cloud computing.*

*The implications of **resource pooling** using **virtualisation** are expected to be strong. This is because the computing resources can be distributed to different hosting sites that may be located in different countries with possibly different applicable legislation. Additionally, the introduced abstraction of resource virtualisation requires additional effort to monitor and report on resource locations. To cover this, a detailed analysis of the technical implications of distributed computing is provided in Section 2.3.2. For virtualisation a detailed analysis is provided in Section 2.3.3.*

*The implications of **rapid elasticity** and **scalability** are limited to usability issues and compliance with SLAs. Therefore, these characteristics are revisited in the evaluation in Section 6.2.2. The **on-demand self-service** and **pay-per-use utility models** relate only to the business contract between cloud provider and cloud customer and have no impact on the legal compliance investigated in this thesis.*

*The **measured service** has strong implications for possible documentation, monitoring, and reporting activities that can be utilised to support compliance management. A detailed analysis of the documentation, monitoring, and reporting capabilities is presented in Section 4.3.*

*The **broad network access** directly refers to the needs of access and transmission control (cf. Sections 3.6.3.2 and 3.6.3.3), since the connection used to access or transfer data can be used maliciously by an attacker (e.g., by spoofing the client's identity or eavesdropping on the communication). The technical abilities to secure communication and to impleemnt access and transfer control are investigated in Section 4.2.3.*

2.1.2 Concepts of utilisation

The utilisation of cloud computing can be distinguished by three service models and four deployment models [141]. All models describe a different form of cloud utilisation, which implementations all have different technical implications on legal compliance of data processing in the cloud. All models and the technical implications are investigated in the following.

2.1.2.1 Service models

The service models describe the type of service that is provided to the cloud customer. Each service model defines a different level of service abstraction representing the administrative effort and flexibility a cloud customer has.

Software-as-a-Service (SaaS) represents the provisioning of *applications* hosted on a cloud infrastructure. The applications are accessible via network using heterogeneous end systems. While the application is running in the cloud, it appears to be running on the cloud customer's end system. SaaS provides the highest level of resource abstraction. The cloud customer has no direct interactions with either hosting infrastructure or the remote operating system. The application integrates seamlessly into the environment on the cloud customer's end system.

Platform-as-a-Service (PaaS) describes the provisioning of a *platform* for development and deployment of applications hosted on a cloud infrastructure. The platform provides the cloud customer with a framework with which to design, implement and deploy cloud applications. Such a framework may provide software development tools as well as libraries and services that can be included in the developed applications. As with SaaS, the framework is accessible via network using heterogeneous end systems. The level of abstraction is high, since the cloud customer has no direct interaction with either the hosting infrastructure or the remote operating system. However, the development and deployment of cloud applications requires access to the basic functionality of deployment mechanisms, which leads to an abstraction level lower than in SaaS, since the deployment and operation of applications are also managed by the cloud customer.

Infrastructure-as-a-Service (IaaS) represents the provisioning of *computing resources* in the form of freely configurable virtual resources (e.g., virtual machines and storage) that are

hosted on a cloud infrastructure. The cloud customer fully controls the software running on the virtual resources (including operating system and applications). The cloud provider may offer predefined installations of operating systems and applications to the cloud customer. The level of abstraction is low, since the cloud customer has full access to the virtual resources and has to manage software deployment and operation on his or her own. However, the underlying cloud infrastructure is not visible to the cloud customer and the provisioning of computing resources is fully under control of the cloud provider.

Remark 2.3 (Technical implications of service models) *All three service models have in common that they have an abstract view on the provisioned computing resources. The computing resources are created by resource virtualisiation enabling their management and provisioning by resource pooling. As mentioned in Remark 2.2, resource pooling and virtualisation have a strong impact on applicable legislation and on the ability of monitoring and reporting legal compliance, which is discussed in Section 2.3.*

Remark 2.4 (Reduction to IaaS) *An interesting observation is that it is possible in principle to host the platform used for PaaS and the applications used for SaaS on computing resources provided in IaaS. This allows (in a first step) to reduce the inspection of all service models to the inspection of IaaS. Consequently, the analysis of the technical implications of legally compliant cloud computing can also be reduced to the investigation of IaaS. However, it is still necessary to investigate the implications only given by the special characteristics of PaaS and SaaS that are not addressable on IaaS level (e.g., application deployment and application behaviour), but this can be done separately.*

In the following, the technical analysis is reduced to the implications of IaaS. An outlook on possible investigations of PaaS and SaaS characteristics is given in Section 7.3.

2.1.2.2 Deployment models

The deployment models describe the deployment configuration of the underlying cloud infrastructure. Each deployment model defines to what degree the cloud infrastructure is shared among cloud customers.

Private cloud In the private-cloud model, the cloud infrastructure is exclusively provisioned to and used by a single organisational entity (i.e., cloud customer with multiple user accounts). The cloud infrastructure can be on the premises of the cloud customer or off the premises (located at a third-party cloud provider). The responsibility for operation and management of the cloud infrastructure can lie with the cloud customer or the cloud provider, where shared responsibilities are also possible. For the private cloud the level of isolation is very high, since the cloud infrastructure is only accessible by a single cloud customer. The deployment model is very similar to traditional IT outsourcing (cf. Section 2.1.3.3). Therefore the complexity of the shared operation is low, since it can be distinguished by the responsibilities of the cloud customer and the cloud provider.

Community cloud The community-cloud model is similar to the private-cloud model, but the infrastructure is accessible by a group of cloud customers with shared concerns. The responsibility for the operation and management of the cloud infrastructure can lie with one or more of the cloud customers or with a third-party cloud provider, and shared responsibilities are also possible. The level of isolation is still high for community clouds, since the access is limited to a group of cloud customers. However, the complexity of the shared operation can be high, since a large number of cloud customers can be involved in the operation of the cloud infrastructure (and therefore, the interdependencies of the involved components can be complex).

Public cloud In the public-cloud model, the cloud infrastructure is provided by a cloud provider for public usage. All cloud resources are openly usable by any cloud customer. The level of isolation is low, since the resources are shared openly. However, there can be a multi-tenancy model implemented for controlling access to specific cloud resources (cf. Broad network access [141]). The complexity of the operation is low, since there is a single third-party cloud provider operating the cloud infrastructure. In federated cloud scenarios, the complexity of operation can increase with the number of involved cloud providers. Such scenarios are particularly covered by the hybrid-cloud model.

Hybrid cloud It is possible to combine two or more of the previous models to form a hybrid-cloud model. For example, a federated public-cloud infrastructure can be considered a hybrid cloud, since it consists of a number of public-cloud infrastructures that are interconnected via a community cloud belonging to the cloud provider. An other example is a combination of public and private clouds, where the cloud infrastructure can provide openly usable computing resources as well as protected computing resources that are exclusively accessible by specific cloud customers. Depending on the combination of the existing deployment models, the level of isolation and the complexity of operation can be different. As a rule of thumb, the level of isolation is less or equal to what it is in the combined deployment models, and the complexity of operation is greater or equivalent.

Remark 2.5 (National vs. global cloud) *Cloud infrastructures can be located within a single country (**national cloud**) or be distributed among several countries (**global cloud**). This is a novel criterion with which to classify clouds which is covered neither by the service models nor by the deployment models. For that reason, the classification in* national clouds *and* global clouds *is introduced by this thesis explicitly.*

From a technical perspective, there is no difference between a cloud infrastructure is located within a single country an one located in multiple countries. From a legal perspective, this difference is crucial for the legally compliant operation of a cloud infrastructure, since the applicable legislation depends on the country the cloud infrastructure is located in. The most complex case is the operation of a cloud infrastructure distributed globally (i.e., over multiple countries) providing public and private cloud services. Without loss of generality, global clouds are investigated in this thesis. An analysis of the distribution characteristics of cloud infrastructures is provided in Section 2.3.2.

2.1.3 Alternative approaches

The characteristics of cloud computing make it a promising candidate for IT outsourcing in practice. In particular, IaaS and private clouds offer service characteristics that are very similar to traditional IT outsourcing scenarios. However, cloud computing is not the only approach to provide IT services to customers. Examples of existing alternative approaches are examined in the following.

2.1.3.1 Traditional IT outsourcing

Traditional IT outsourcing can be defined as the provisioning of human and/or physical resources by a third-party organisation as a contribution to or replacement for the IT infrastructure of the outsourcing organisation [135]. Following this definition, IT outsourcing can be characterised into five[1] modes: data processing, systems integration, systems design/planning, telecoms/network, and application development. Traditional IT outsourcing is limited in its flexibility, since it is contracted by project or period (cf. [135]). Thus, all resource requirements have to be defined beforehand, including all expected requirements for scalability and utilisation.

In comparison, all modes of IT outsourcing can be addressed by cloud computing. Data processing, telecoms/network, and systems integration can be covered by IaaS and SaaS, and systems design/planning and application development can be supported by PaaS. On the one hand, cloud computing is limited to providing the physical resources (i.e., IT infrastructure), and human resources are usually[2] not provided. On the other hand, cloud computing goes beyond traditional IT outsourcing by providing more flexibility (e.g., on-demand and scalability).

The implications of traditional IT outsourcing on legal compliance can be considered being similar to those of private cloud computing. In both cases, there is a bilateral relationship – established between the outsourcing-service provider (agent) and the outsourcing customer (principal) – covering contracted resource provisioning and utilisation. In particular, the location of the computing resources and the responsibilities associated with operating them can be clearly distinguished by the outsourcing customer and the outsourcing provider.

2.1.3.2 Grid computing

A paradigm closely related to cloud computing is the grid computing paradigm. Both paradigms have much in common their architecture and technology (e.g., large scale, distribution, resource pooling, remote access) [24], but there are also differences, for example, with respect to computation and data model, application, and abstraction [74]. The service and deployment models of grid computing are different from that of cloud computing, but with respect to legal compliance, they both share the same issues arising from resource virtualisation and distributed

[1] Loh and Venkatraman [135] also mentioned 'data centre' as a sixth mode of IT outsourcing. In today's understanding of outsourced data processing, the outsourced operation of a data centre can be considered a large-scale application of outsourced data processing, and can therefore be merged with 'data processing'.

[2] There are approaches to utilising human resources in cloud computing, e.g., Amazon Mechanical Turk, on the Internet: http://aws.amazon.com/de/mturk/ (last visited: 30.06.2015).

computing. Simplified, grid computing can be considered to be similar to IaaS, while having supercomputing capabilities instead.

2.1.3.3 Other distributed service-oriented architectures (SOAs)

There are other approaches to providing distributed computing services. Prominent examples are Web Services, which are defined by the Web Service Description Language (WSDL) [220], and Common Object Request Broker Architecture (CORBA) [154]. Both provide middle-ware for machine-to-machine (M2M) communication utilising remote procedure calls (RPCs). Unlike cloud computing, the mentioned SOAs do not provide computing resources, but are used to exchange data or perform operations on provided data. However, the endpoints of the RPC can be globally distributed, and the introduction of middle-ware for communication introduces an abstract view on the endpoints similar to the resource virtualisation in cloud computing. Consequently, cloud computing and distributed SOAs have similar challenges with respect to legal compliance, although they are implemented using different technologies.

2.2 Understanding IT outsourcing to cloud infrastructures

Within the scenario of IT outsourcing to the cloud, the cloud customer contracts the cloud provider to provision cloud resources that are utilised to operate the cloud customer's IT infrastructure. Beside the cloud provider and the cloud customer, there are additional actors that are also involved. For example, the hardware resources of the cloud infrastructure are operated by one or more hosting sites. It is important for the achievement of legal compliance that the information flow between the actors involved should comply with the requirements of applicable legislation and contractual agreements. If information flow complies with applicable legislation and contractual agreements, the information flow is considered *allowed*. Otherwise, the information flow is considered *forbidden*. Further, if there is allowed information flow between actors, these actors are considered *legitimate*, and otherwise *illegitimate*. Inversely, all information flow to an illegitimate actor is considered forbidden. An example of an illegitimate actor is an attacker.

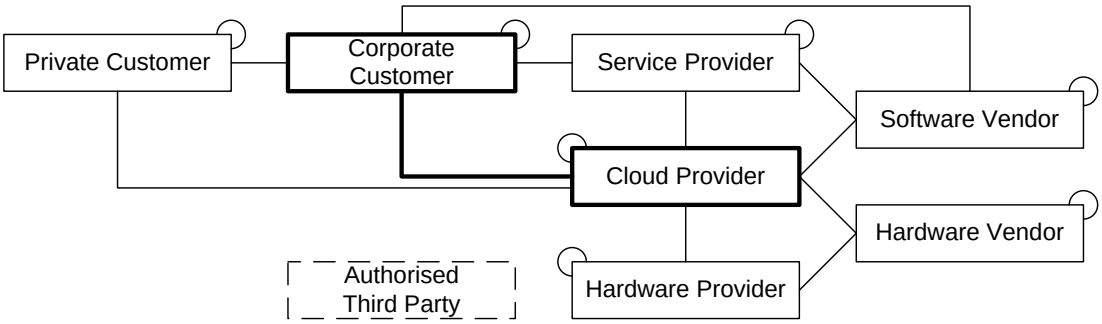
Even though there is allowed information flow between legitimate actors, there might be information that is not allowed to be shared with all legitimate actors. For example, information that is allowed to be processed only within a specific country (e.g., German tax data), is allowed to flow only between legitimate actors located in that specific country. Therefore, it is necessary to differentiate between allowed and forbidden information flow with respect to legitimate actors.

Section 2.2.1 identifies the possible legitimate actors that are involved in IT outsourcing to the cloud, and, based on their interaction, a relationship model is defined, which is used in this thesis. In Section 2.2.2, the relevance of legal compliance for the key actors, i.e., cloud provider and corporate customer, is investigated. Further, basic reasons and incentives for implementing legal compliance in clouds are identified.

2.2.1 Legitimate actors and their interaction

In this thesis, the cloud model consists of seven different actors that can be involved in an IT outsourcing scenario. Figure 2.1 depicts these actors and the way they interact. In any given instance, each actor can be absent or represented one or multiple times. The interaction between actors is bidirectional and can be one-to-one, one-to-many, and many-to-many. The actors interact with their connected neighbours and in particular with instances of themselves.

Figure 2.1: Legitimate actors and their relationship in IT outsourcing to cloud infrastructures.



The main interacting actors are the corporate customer and the cloud provider, but there are also private customers, the service provider, the hardware provider, the software vendor, the hardware vendor, and authorised third parties. Authorised third parties like governmental authorities (e.g., tax authority) and assigned inspectors (e.g., certified public accountant) are considered to be a special case, since these actors are not involved in the outsource scenario itself, but can be involved when executing a legal act (e.g., tax inspection) or an audit. In such a case, the authority might interact with any of the involved actors as necessary.

Each of the legitimate actors is described in the following.

Cloud provider The cloud provider is considered the actor operating the cloud infrastructure as a whole. A cloud provider can be responsible for operating the cloud infrastructure within a single country (i.e., operating a national cloud or a local cluster of a global cloud), but also can be responsible for operating the cloud infrastructure in multiple countries (i.e., operating a global cloud or a global cluster of a global cloud). The cloud provider is responsible for all operations performed by the cloud infrastructure and, in particular, for the cloud's management. The cloud provider interacts with the corporate customers using cloud services for IT outsourcing and with the service provider offering services on the cloud platform, and with the hardware provider subcontracted to operate the hosting sites. Additionally, the cloud provider interacts with vendors with regard to software and hardware which are used for the cloud offers. In the scenario of federated cloud computing, cloud providers interact with each other when they are exchanging data or cloud resources. This is also true when local or global clusters of a global cloud are operated by different cloud providers.

Service provider A service provider is an actor operating and managing cloud resources without operating a cloud infrastructure. Instead, the cloud resources of a service provider are hosted on a cloud infrastructure operated by a cloud provider. A cloud provider can be a service provider, too. In particular, this is the case when a cloud provider manages and operates cloud services in a global cloud where global or local clusters are operated by individual cloud providers. The service provider interacts with the corporate customers using the cloud services for IT outsourcing, with the cloud providers hosting the cloud services, and with the software vendors supplying the software used in services. Additionally, service providers interact with each other when using other cloud services for providing their own cloud services (e.g., using a spam filtering service for an email service). Service providers are responsible for the cloud services they offer. The subcontracted cloud provider acts on behalf of service providers.

Hardware provider A site that operates hardware resources which is utilised to host cloud resources for a cloud provider is considered a hardware provider (e.g., hosting sites and data centres). A hardware provider can also be a cloud provider when hardware resources and cloud resources are both operated by the same provider. Operating the hardware resource of the cloud infrastructure, the hardware provider is primarily interacting with the cloud provider and also with other hardware providers when exchanging data or cloud resources with other hardware providers. A hardware provider also interacts with hardware vendors individually with respect to obtained hardware. The hardware provider is responsible for hosting cloud resources as requested by the cloud provider and acts on behalf of the cloud provider.

Software vendor Software vendors license and/or sell the software used in cloud services. This licensing/selling interaction is done with the cloud provider, the service provider and the corporate customer, depending on who is using the software and the specific licence model. Usually, licensing is done before software is used. Therefore, software vendors have only little influence on the actual processing of data within the cloud. However, the quality and logic of the software have an influence on the security of data processing within the cloud. A software vendor might be interacting with other software vendors when utilising third party software for their own products. Depending on the licence model, the software vendor can be responsible for security incidents and errors caused by the software.

Hardware vendor The hardware vendor leases or sells the hardware used by the hardware provider. Therefore, the hardware vendor interacts with the hardware provider directly or with the cloud provider in case that the cloud provider buys the hardware which is then operated (on behalf of the cloud provider) by the hardware provider. Further, the hardware vendor interacts with other hardware vendors when utilising third party hardware for their own hardware. Usually, the hardware provider is responsible for the reliability of the hardware, but not for the security of the software and cloud resources that are hosted on it. In detail, this depends on the contract between the hardware vendor and the hardware provider (or the cloud provider).

Corporate customer Corporate customers are considered organisations that outsource their IT to the cloud or use cloud services. In the context of IT outsourcing there are only corporate

customers using cloud resources, since there are no private customers using the cloud. Therefore, corporate customers interact on the one hand with their customers, which might be private persons or other organisations. The latter are considered corporate customers, too. On the other hand, corporate customers interact with the service and cloud providers directly in order to access and manage their cloud services. They also might interact with software vendors for the purpose of licensing or acquiring software that is used within the cloud resources. Corporate customers are responsible for the data processing done on cloud resources.

Private customer Private persons whose personal data is processed within the cloud are considered private customers. In the scenario of IT outsourcing, private customers do not use cloud services on their own. Nonetheless, private customers get involved when their interact with a corporate customer (i.e., if they are the customer of the corporate customer). In this context, private data might be processed on behalf of corporate customers within the cloud if corporate customers move data belonging to their private customers into the cloud. The corporate customer is responsible for doing this, but the cloud provider can also become responsible for processing private data in the cloud, for example, when transmitting data to the outside of the EU/EEA (cf. Section 3.2). Usually, private customers interact only with corporate customers. Private customers may also interact with each other, for example, when sharing data with each other or interacting as a group. If data are transmitted outside of the EU/EEA, there might be interaction with the cloud provider (for example, when the private customer claims the right of access).

Authorised third party Authorised third parties are considered all entities that are not involved in the IT outsourcing directly but get involved due to a legal authorisation or an assignment by one or more of the involved actors. Examples of such third parties are governmental authorities (such as the tax authority according to German tax law; cf. 3.4.2), data protection authorities according to European data protection law (cf. Art. 20 No. 2 of the Directive 95/46/EC), and assigned inspectors performing an audit (such as the certified public accountant in conjunction with the standards on service organisation controls SSAE 16 and the ISAE 3402). These third parties have in common that they are not involved in the process of IT outsourcing but authorised (by law or contractual agreement) to verify compliance with legal and/or contractual requirements. Therefore, they have to interact with all actors that are under inspection and, particularly, review the data processing performed by these actors. An example is the inspection of a German cloud customer by the German tax authority. For such an inspection, it is necessary to review the data processing systems including those systems used for outsourced data processing (according to GDPdU[1] No. I.1.a). In this example, the inspection of the corporate customer by the tax authority includes the inspection of related data processing at all involved cloud providers' and hardware providers' sites.

In general, authorised third parties might interact with any other involved actors depending on the specific case and granted authority. In the following, it is assumed that authorised third parties interact with all other involved actors. This rather general assumption makes it

[1] Administrative regulation "Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen (GDPdU)" (in German) established by the German Federal Ministry of Finance (cf. BMF IV D 2 - S 0316 - 136/01).

possible to focus the information model on the information flow between the actors originally involved in the IT outsourcing scenario without neglecting the possible information flow to the authorised third parties. However, it might be necessary to control the information flow to authorised third parties more specifically (e.g., tax authorities must have access to tax data but usually not necessarily to personal data). This requires additional research with a focus on information flow to authorised third parties, which is not part of this thesis. Possible extensions of the model that is presented in this thesis are discussed in the outlook (cf. Section 7.3).

2.2.2 Relevance of legal compliance

Having identified cloud providers and corporate customers, the key actors in the scenario of IT outsourcing to clouds, it is necessary to understand the meaning of legal compliance from the perspective of these actors. Their reasons and incentives for implementing legal compliance form the basis of investigating legal requirements and their implementation in the context of IT outsourcing to clouds.

Corporate customers have to comply with legislation generally. For example, they have to comply with data protection law when processing personal data, and with tax law when processing data relevant to tax law. These legal requirements also apply to corporate customers when they are outsourcing their IT to clouds. Additionally, restrictions may apply to the outsourcing itself, for example in German tax law outsourcing is allowed only with explicit permission of the competent revenue authority (§146 para. 2 cl. 1 AO). When outsourcing data processing, the corporate customers are usually responsible for this act (e.g., for personal data in EU/EEA according to Art. 7 lit. a in association with Art. 6 para. 1 Data Protection Directive). Therefore, it is in their interest that the service organisation (here the cloud provider) should act in compliance with legal constraints applicable to the corporate customers. They may even be obliged to inspect the cloud provider (e.g., according to German data protection law, §11 para. 2 cl. 4 BDSG). For that reason, the technical implementation of clouds has to support the inspection of the cloud provider by the corporate customer. In this context, it is of particular interest, which characteristics and requirements are covered by such inspections and therefore can be monitored and reported on.

Cloud providers are contracted by corporate customers and have to comply with the bilateral contract including SLAs addressing legal constraints of the corporate customers. For example, if the corporate customer is obliged to guarantee the accessibility of their tax data to the competent revenue authority, the cloud provider has to ensure the availability of the tax data in a case of an inspection by the competent revenue authority. For the cloud providers, it is important to know what legal requirements of their corporate customers they have to fulfil upon service delivery. Cloud infrastructures have to be capable to support these legal requirements by decision and enforcement mechanisms as well as monitoring and reporting mechanisms. Due to the high degree of automation, these mechanisms have to operate with little or no human interaction if possible. However, inspections by the cloud provider, corporate customers, and competent authorities must be possible. Further, cloud providers usually subcontract hardware providers and third party hardware provider, which also have to comply with the legal requirements of the corporate customers. To achieve this, decision-making and enforcement mechanisms as well as monitoring and reporting mechanisms have to cover the subcontracted

provider as well as the cloud provider.

In this thesis, these requirements are addressed in a first step by identifying legal requirements of the corporate customers that imply requirements on technical implementation at the cloud provider. This is covered by the legal analysis in Chapter 3. In a second step, technical ability of clouds to cover identified legal requirements is investigated in Chapter 4. The identified research gap is covered in a final step, performed in Chapter 5, by an information model addressing the decision and enforcement of legal requirements in clouds, which also supports monitoring and reporting mechanisms.

2.3 Technical impact of clouds on legal compliance

In Section 2.1 several technical observations that are important for investigating the legal compliance of cloud infrastructures were made. First, cloud infrastructure consists of *virtual resources* and *hardware resources*. To evaluate the legal compliance of cloud infrastructures it is necessary to understand their relevance. Second, the utilisation of *distributed computing* and *virtualisation* in cloud computing are important characteristics of cloud computing, having a severe impact on the legal compliance of cloud infrastructures. Their technical implications when it comes to legal compliance have to be clarified. In the following, the technical implications of cloud infrastructures, distributed computing, and virtualisation of the evaluation process and the legal compliance in general are discussed.

2.3.1 Evaluating cloud infrastructures

Cloud computing can be described as the process of mapping virtual resources to hardware resources. To evaluate the legal compliance of a cloud infrastructure, it is necessary to investigate legal compliance on the level of virtual resources, hardware resources and the cloud management process.

While the legally compliant utilisation of virtual resources is the cloud customer's responsibility, the provisioning and hosting of virtual resources is the cloud provider's responsibility. The provisioning is made at the cloud customer's request. Therefore, the cloud provider has to ensure compliance with the cloud customer's preferences, which are usually regulated by SLAs that are expressed via the cloud customer's management front-end and enforced by the cloud management process. Thus, the legal compliance of provisioning and hosting virtual resources depends on the legally compliant execution of the cloud management process.

The cloud management process assigns hardware resources for the operation of virtual resources. Its legal compliance depends on assigning hardware resources that legally comply with the requirements of the virtual resources (i.e., the cloud customer's preferences).

To conclude, the legal compliance of the cloud infrastructure depends on the legally compliant operation of the underlying hardware resources and their assignment to requested virtual resources. The technical capacity of the cloud provider to enforce legally compliant operation and assignment depends on the implemented cloud security management, which is discussed in Section 4.2. The impact of distributed hardware resources and their abstraction to virtual resources is discussed in Sections 2.3.2 and 2.3.3.

2.3.2 Distributed computing and location inhomogeneity

The abstract view on the underlying hardware resources allows the hosting of virtual resources at different locations seamlessly. The decision where virtual resources are hosted is done on a macro level by selecting the hosting site and on a micro level by selecting the hardware resource at the hosting site (cf. Section 4.1.4). On both levels, the decision affects the legal compliance of virtual resources.

On the **macro level**, the selection of the hosting sites influences which country the virtual resources are effectively hosted in and who is locally responsible for the operation of the hosting hardware. Hosting sites can be distributed over different countries, and therefore, they can be located in different jurisdictions. Therefore, selecting the hosting site implies selecting the applicable legislation, which is key to the make legal compliant decisions in respect to the location. Furthermore, the hosting sites are usually operated by different organisations and responsible persons. This means that the responsibilities for operating the hosting hardware are split among a group of entities (i.e., private or legal persons), which can individually implement security and compliance management. This can result in a technical and organisational divergence on effective security and compliance between hosting sites. To some extent, this divergence can be compensated for by the cloud provider (e.g., by communicating unified security policies and monitoring the effective security at the hosting sites). For flexible and robust security management, such divergences have to be considered when making decisions on allocating resources to specific hosting sites. Therefore, specific information on the current state of security and compliance at each hosting site has to be communicated to the cloud provider. On the other hand, the hosting sites have to be informed of security policies and applicable constraints of the allocated cloud services. The latter information is particularly important for the enforcement of applicable requirements at the hosting site.

On the **micro level**, the cloud management process decides which hardware is locally used to host virtual resources. In general, the hardware equipment at a hosting site is not necessarily homogeneous, because hardware tends to be stocked or substituted successively rather than replaced completely on every upgrade or replacement. This consequently results in a heterogeneous mix of servers at most hosting sites with differences in hardware-supported security features. On the one hand, this heterogeneity can be intentional to provide the ability to cover different hardware requirements of virtual resources (e.g., GPU support). On the other hand, it can result in different sets of security measures implemented. For example, servers with the Trusted Platform Module (TPM) installed support the operation of trusted virtual machine monitors (TVMM) [79], which provide mechanisms to ensure and validate the integrity of executed software. However, it cannot be assumed that every server has TPMsupport installed (e.g., the FUJITSU Server PRIMERGY Blade-Systems do not support TPM by default[1]). For the compliant operation of virtual resources, the differences in local hardware when it comes to security features has to be be considered in the cloud management process.

As discussed above, the distributed operation of cloud services requires the cloud management to deal with inhomogeneity in several ways. In this context, location is an important

[1]According to the data sheets on the Internet http://globalsp.ts.fujitsu.com/dmsp/Publications/public/ds-py-bx400-s1.pdf and http://globalsp.ts.fujitsu.com/dmsp/Publications/public/ds-py-bx900-s2.pdf (Last visited: 30.06.2015)

property to address and manage these inhomogeneities, which can be definedas follows.

Definition 2.2 (Location in-/homogeneity) *A cloud infrastructure has the property of* ***location inhomogeneity*** *if and only if its underlying hardware resources are distributed among different hosting sites which (a) are located in different countries (with different applicable legislation) or (b) implement different sets of security measures.*

Conversely, a cloud infrastructure has the property of ***location homogeneity*** *if and only if the hardware resources are distributed among hosting sites which are located in a single country (or in different countries with the same applicable legislation) and implement the same set of security measures.*

Remark 2.6 (Distinction from location diversity) *The definition of location inhomogeneity has to be distinguished from the definition for* ***location diversity*** *used in the anonymity context of location-based services. There, location diversity defines a measure for the probability of associating a communication process (i.e., query) with a certain location [222]. Both definitions have in common that they define a property that is resulting from interpreting the data processing at multiple locations. Strong location diversity means that there is a large number of undistinguishable positions, because they have the same observed properties. This is diametrically opposed to location inhomogeneity, where the observed properties are necessarily distinguishable (by jurisdiction or responsible operator). However, location homogeneity supports location diversity, since the jurisdiction and the responsible operator are the same for all positions of data processing.*

Remark 2.7 (Hardware in-/homogeneity) *To address the difference in the location of supported computational and security features independently, such a difference is called* ***hardware inhomogeneity****. On the other hand,* ***hardware homogeneity*** *expresses the fact that there are no differences in the supported computational and security features. Hardware in-/homogeneity can be used to describe the hardware resources of cloud infrastructures, of global/local clusters, and of hosting sites.*

2.3.3 Impact of virtualisation

The main implication of virtualisation is that the hardware resources are hidden from the cloud customer and can be easily exchanged on demand. While this obfuscation and flexibility is an intended feature of the cloud management process, it leads to new challenges in monitoring and validating the effective legal compliance. In general, there is no dedicated hardware in the cloud that can be monitored and validated by cloud customers with regard to its legally compliant operation. This is true only for private cloud scenarios. In particular, the cloud customer usually[1] has no contact with the hardware operators – in contrast to traditional outsourcing scenarios (cf. Section 2.1.3.1) – to gather evidence on the effective legal compliance of the hardware resources. Without this knowledge, cloud customers have to rely on information provided by the cloud provider.

[1] In general, the cloud customer has contact with the cloud provider, which subcontracts the hardware sites. An exception from this is when the cloud provider is also the hardware provider.

While the possibilities had by the cloud customer to monitor and validate the effective legal compliance are limited, the customer's services can be arbitrarily distributed on different hosting sites and be co-hosted with services of other customers without the customer's notice. This can result in an unnoticed breach of required security constraints, for example, in illegitimated transfer of German tax data to a third country (cf. Section 3.5.3.2) or unauthorised disclosure to other cloud customers.

To conclude, the compliance control of the IT services outsourced by the corporate customer depends on the reports and evidence provided by the cloud provider. The abilities of cloud providers to document, monitor and report on compliance are discussed in Section 4.3.

2.4 Summary of related work

This thesis investigates the mutual dependencies of legislation and technology in the context of IT outsourcing to clouds with respect to European and particularly German legislation. To address open challenges, an information theoretical implementation of location-determined data processing is proposed that deals with location inhomogeneity (cf. Def. 2.2) in clouds and supports legal compliance.[1] Dealing with the location of data processing has been identified as a legal and technical challenge by multiple authors (e.g., in terms of 'data location' [117] [41], 'data locality' [191], and 'multi location issue' [230]). However, a detailed analysis of literature in respect to this particular topic (cf. Section 4.2) revealed that location inhomogeneity in clouds is only partly addressed by current research.

It is possible to decide and enforce the manner in which virtual resources are assigned to hardware resources in clouds based on countries using a list of secure third countries in the context of European data protection law [202]. Further, the virtual machine placement can be attested remotely in relation to the hosting hypervisor and compute server using technology of trusted hypervisors and trusted platform modules [180, pp. 12 et seqq.]. The same is true for cloud storage [3] [4]. When tagging clouds resources with information on their (geo-)location, cloud providers can compliantly assign virtual resources to hardware resources by location constraints [217]. There also exist multiple approaches to resource allocation strategies considering location constraints (e.g., location-aware MapReduce [122] [161] [162] [92], customised location constraints [128], co-location of virtual machines [14], optimal location in cloud networks [129], customer-centric resource allocation [187] [196], and inter-cloud architectures [90]). However, if legal aspects are considered, only data protection law is considered. Other legal requirements such as those of given by tax law or export control are not considered. Further, none of the mentioned approaches covers an information model which verifies and ensures the correctness of the decisions and enforcements (in respect to assignment of virtual resources to hardware resources) made by the cloud providers.

Related work on information flow control is investigated in Section 5.2.1 and Section 5.2.2, which reveals that methods of MAC are applicable, but the location inhomogeneity in clouds has not been addressed, yet. Applying information flow control to clouds has been considered before. Having the development of secure cloud services in mind, Bacon et al. propose to use

[1] In Chapter 4, this problem is defined more specifically as the *challenge of location inhomogeneity* (cf. Def. 4.6).

methods of information flow control in PaaS cloud environments [13]. Based on the groundbreaking approaches of Bell and La Padula [20], Denning [52], Biba [22], it is possible to prove confidentiality and integrity with respect to information flow in systems generally (including IaaS cloud environments like shown in Section 5.2.3). Further, these models can be combined into a single model addressing both confidentiality and integrity [175]. It is also possible to consider location constraints such as co-location [201] and user location [166]. However, current research does not cover the modelling of location constraints, which apply when assigning virtual resources to hardware resources and which are based on the legal requirements of corporate customers. Also, none of the current approaches supports availability, which is an important legal requirement, e.g. in the context of German tax law (cf. Section 3.3.3).

To conclude, the review of current research revealed that there is awareness of location inhomogeneity and its importance for legally compliant cloud computing, but that current approaches focus on data protection law (other legal norms are not considered) or do not consider legal requirements at all. Further, there exist multiple approaches addressing location constraints, when assigning resources in clouds. However, none of them supports the reliable verification of decisions made and enforcements when it comes to the information that is allowed flow between hardware resources. With the methods of information flow control, the decision on and enforcement of allowed information flows can be verified with respect to confidentiality, integrity and co-location of virtual resources. Nonetheless, availability and the location of hardware resources are not supported by existing approaches. This thesis provides a novel approach, extending methods of information flow control to cover availability and location constraints in order to ensure legally compliant cloud resource allocation based on legal and technical requirements in IT outsourcing to clouds.

2.5 Conclusions on legally compliant clouds

When corporate customers are outsourcing their IT to IaaS clouds, it is important that the provisioning and hosting of the utilised virtual resources should be legally compliant with applicable legislation. Which legal requirements apply can be different for each individual corporate customer and is investigated in Chapter 3. The cloud providers have to translate these requirements into legally compliant cloud computing. While cloud computing consists of hosting, provisioning and utilisation of computing resources, this implies three different views on computing resources: 1) the physical view on the hosting infrastructure (i.e., **hardware resources**), 2) the logical view on resource provisioning (i.e., **cloud management**), and 3) the logical view on resource utilisation (i.e., **virtual resources**). All views have a different level of abstraction, including management and utilisation of cloud resources. To ensure legally compliant data processing in clouds, it is necessary to understand the technical mechanisms of hosting, provisioning, and utilisation and to identify their impact on legal compliance. In particular, the technical concepts behind resource pooling and virtualisation are key to this understanding. In this thesis, these concepts and their interconnection are investigated for IaaS (cf. Remark 2.4) in hybrid clouds which are globally distributed (cf. Remark 2.5). The capacity of cloud environments to support legal requirements in such a scenario of IT outsourcing is investigated in the technical analysis in Chapter 4. The identified research gap is then addressed

by proposed methods of information flow control in Chapter 5, which is evaluated in Chapter 6. Chapter 3 investigates the underlying legal requirements of IT outsourcing to clouds.

Chapter 3

Legal analysis and technical requirements

This thesis aims to identify and implement the legal requirements of outsourced data processing in clouds technically.[1] A particular focus lies on data and corporate customers originating both from EU/EEA countries in general and particularly Germany, while the cloud providers may operate globally. In such a context, the challenge arises that data which are protected by European and German legislation may be processed in foreign jurisdictions. Legislation of the origin and foreign country may not be compatible, resulting in conflicting requirements,[2] or the foreign jurisdiction lacks adequate regulations.[3]

This chapter identifies the legal requirements for IT outsourcing using cloud computing in the specific case of European and German legislation (Objective 1). Particularly, the legal requirements that a cloud provider has to satisfy – and therefore to implement technically – are investigated with the focus on data protection law and selected sectoral requirements commonly found in the context of IT outsourcing. A specific focus is on identifying legal requirements which have an impact on the technical implementation of clouds. For this purpose, it is necessary to identify corresponding legal norms on IT outsourcing and data processing of European and German legislation and clarify the lawfulness of cloud computing. Based on corresponding legal norms, it is possible to understand the technical implications and derive general and specific requirements for implementing clouds.

In Section 3.1, the lawfulness of cloud computing is examined based on the data types processed in the cloud. In particular, legal norms regulating data processing and basic requirements are identified. Section 3.2 investigates the specific requirements for processing personal data in the cloud using the example of *carrying out data processing* under Germany data pro-

[1] Parts of the structure and content of this chapter trace back to preliminary work which were performed in the context of an expert opinion given to Fujitsu Technology Solutions GmbH. The expert opinion has not been published.

[2] For example, the USA PATRIOT Act obliges European companies affiliated with companies located in the United States of America (USA) to provide access to customer data to investigative authority without consent nor notice of the data subjects, which is generally conflicted with European data protection law [219].

[3] For example, in the context of transmitting personal data to third countries, third countries may lack of an *adequate level of protection* (recital 57 in conj. with Art. 25 para. 4 Data Protection Directive).

tection law. In particular, the responsibility and duties of cloud providers and restrictions on data transmission are derived. Section 3.3 analyses requirements on necessary safeguards at the cloud provider. Especially technical safeguards are investigated, including requirements for implementing confidentiality, integrity, and availability. In Section 3.4, the sectoral requirements of data processing in clouds are examined. For exemple, the specific requirements in the financial, medical and healthcare, and public sector as well as those prescribed by tax law and regulations on export control are covered. In Section 3.5, special requirements that apply with particular relevance for the context of IT outsourcing – and therefore require specific care by the cloud provider – are discussed. Particularly, requirements for the storage of data, documentation in the cloud, special access by investigative authorities, and location-determined data processing are examined. Section 3.6 concludes with the technical requirements implied by the legal requirements investigated in this chapter.

3.1 Lawfulness of cloud computing

In principle, the lawfulness of data processing is regulated by legal norms that are applicable to the specific context of data processing. For example, data protection law regulates the processing of personal data, and tax law regulates the processing of tax data. The data processing must follow these legal norms. The lawfulness of cloud computing is given if data processing within the cloud is in compliance with all applicable legal norms. Each legal norm specifies the circumstance under which it is applicable. Further, each legal norm provides basic requirements that have to be obeyed by corporate customers and cloud providers as well.

This section determines the categories of data that are processed in the cloud and the corresponding legal norms. Further, basic requirements are identified which have to be considered when processing these categories of data in the cloud.

3.1.1 Data categories and corresponding legal norms

In the context of IT outsourcing to the cloud, the corporate customer transfers data to the cloud originating from multiple sources and designated for different processing purposes. For example, data may contain customer information, economic data, and intellectual property belonging to third parties. Whether data processing is lawful and which requirements apply is defined by legal norms corresponding to the processed data.

Generally, data can be classified by the following four categories: personal data, business data, property right protected data, and unprotected data. Each category is introduced below, and the corresponding legal norms are identified. Further, these categories are not necessarily disjunct. For example, customer data may be considered personal data since they reference real persons, and at the same time they are business data since they are associated with a specific corporate customer and are of importance for that customer's operational business. If data belong to multiple categories then multiple norms and the requirements of each category involved may apply. Generally, such data are denoted **mixed data**.

Personal data (also *personally identifiable information*[1]) are data that are (or can be) associated (i.e., personally identifiable) with a real person (a.k.a. the *data subject*) in general).[2] Within the cloud, the personal data of corporate customers' private clients or employees can be processed. In the European Union, protection of personal data is a basic right (Art. 8 Charter of Fundamental Rights of the European Union)[3] and protected by the *Data Protection Directive* (Directive 95/46/EC). In future, the Data Protection Directive will be replaced by the upcoming General Data Protection Regulation (GDPR), which is not to be expected to happen before 2017.[4] In this thesis, the investigations will focus on current data protection law but also upcoming changes made by the upcoming regulation are considered where applicable.

The Data Protection Directive is implemented by the national data protection acts of each member state. In Germany it is generally the BDSG which regulates the processing of personal data by public bodies and authorities of the federal government and non-public agencies. Additionally, there exist data protection laws in each federal state regulating data processing by its public bodies and authorities. However, there are special laws regulating data protection within a specific context which have priority over the BDSG (§1 para. 3 cl. 1 BDSG). For example, personal data within electronic communication, i.e., customer data, traffic data, usage data, and content data, are protected by the Telemediengesetz (TMG), primarily implementing the Directive on Electronic Commerce (Directive 2000/31/EG) – the latter explicitly does not cover data protection (Art. 1 para. 5 lit. b Directive on Electronic Commerce)[5] – and the Telekommunikationsgesetz (TKG) implementing in particular the *Directive on Privacy and Electronic Communications* (Directive 2002/58/EC). In electronic communication, the protection of content data (i.e., the confidentiality of the communications, Art. 5 Directive on Privacy

[1] Established by the NIST [140] and originally defined by the United States Government Accountability Office [209]: "any information about an individual [...], including (1) any information that can be used to distinguish or trace an individual's identity [...]; and (2) any other information that is linked or linkable to an individual [...]."

[2] In Germany, there is an ongoing legal discussion about whether personal identifiability is measured by the scope of the controller/processor (i.e., *subjective* scope) or by the general possibility (i.e., *objective* scope) [28, ch. 1 §3 III.1]. On European level, the Article 29 Working Party stated [11, pp. 15 seqq.] that "a mere hypothetical possibility to single out the individual [sic!] is not enough to consider the person as identifiable", that the purpose of data processing is "one relevant factor [...] for assessing all the means likely reasonably [sic!] to be used either by the controller or by any other person" and that a "case-by-case analysis should be carried out". This generally supports the *subjective* scope but does not necessarily exclude the *objective*, particularly not for purposes including admissible transfer to third parties. Then again, cloud computing can provide a large variety of means for identifying individuals including access to additional information sources and data mining tools on demand [164], where the use of these means becomes more likely. In this thesis, the applicability of the large scope (i.e., generally identifiable) is assumed since case-by-case analyses are beyond the scope of this theses and legal certainty is generally provided due to being compliant with the most general case. This is in conformance with the recommendation by Brennscheidt [28, ibid.] and the definition on *linkability* of personally identifiable information by the NIST [140, ch. 2.1].

[3] In the European Economic Area (EEA), this right is established by Art. 8 European Convention on Human Rights which covers privacy generally and particularly data protection in correspondence to the ECtHR's interpretation (decision dated 13.11.12, application no. 24029/07, recital 187).

[4] The regulation has not yet been finalised (June 2015), and will apply two years after coming into force (Art. 91 GDPR). At that point the regulation will become directly applicable (Art. 288 para. 2 TFEU) and replace all national data protection law. Particularly in Germany, many sectoral regulation will become obsolete [104].

[5] This is the reason that the TMG implements the Data Protection Directive only for tele-media services (accommodating the priority of TMG's area specific regulations on data processing over the BDSG), which usually are information society services (according to Directive 2000/31/EC).

and Electronic Communications)[1] and the protection of location data other than traffic data (Art. 9 Directive on Privacy and Electronic Communications) are regulated separately.

Furthermore, in the context of German social security services, personal data are protected by the Sozialgesetzbücher (SGB), and moreover, personal data are protected by the StGB if there are obligations of professional secrecy (§203 StGB in association with Art. 8 para. 3 Data Protection Directive). Additionally, personal data that are given to a German finance authority, i.e., tax data, are protected by tax secrecy regulations according to §30 AO. Brennscheidt mentions that there exists a large number of area specific regulations on data protection in Germany [28, ch. 1 §3 III.4.c].

Encrypted personal data are considered personal data that are encrypted using a secure[2] cipher algorithm. At least in German legislation, it is an open question whether encrypted personal data are still protected by data protection law [28, ch. 1 §3 III.1]. A hypothesis formulated by Spies [188] is that data protection law generally applies on transfer of encrypted personal data into the cloud, unless the cloud provider proves that the data are encrypted and that only the data subject can access the data (with the key necessary for decryption). Spies argues that encrypted personal data may not be considered personal data but anonymised data (according to §3 para. 6 BDSG). Current investigations by Brennscheidt indicates that the discussion is still open but there are strong arguments[3] for considering encrypted personal data still protected by data protection law. At a European level, the *Article 29 Data Protection Working Party* consider "encryption with secret key" *pseudonymisation* which "when used alone will not result in an anonymous dataset" since it "reduces the linkability" and, therefore, is "a useful security measure but not a method of anonymisation" [12, ch. 4]. This is also in conformance with the NIST's definition of the *linkability* of personally identifiable information [140, ch. 2.1]. Consequently, in this thesis, personal data are considered continuously protected by data protection law whether they are encrypted or not.

Business data are data associated with an organisation and usually contain *trade secrets*. In clouds, business data of corporate customers and corporate clients of a corporate customer might be processed as well as business data from software providers, hardware providers, service providers and cloud providers themselves. Business data are usually protected by non-disclosure agreements (NDAs) in business contracts. Particularly the outsourcing contract between cloud provider and corporate customer may contain NDAs. There are additional requirements for data processing that may apply to business data for specific contexts. For example, in Germany, business data are protected against disclosure by §203 StGB if there are obligations of professional secrecy and in the context of unfair competition (§§17, 18 Gesetz gegen den unlauteren Wettbewerb (UWG)).

[1] In Germany, content data are protected by the BDSG and, in the context of telecommunication, also by the secrecy of telecommunication according to §88 TKG.

[2] In the sense of using current cipher algorithms with sufficient key lengths, which are adequate to protection requirements and period of usage according to existing recommendations [85].

[3] According to Brennscheidt [28, ch. 1 §3 III.1]: Encryption is a necessary safeguard which shall not change the data themselves. Further, decryption might be possible in the future without the necessary key due to progress in breaking the cipher. Additionally, the applicability of data protection law may change during processing – due to encryption and decryption – resulting in unpredictably of the effective protection, which is to the data subject's disadvantage.

Property right protected data are data that contain (or represent) intellectual property and, therefore, are protected by intellectual property right and copy right law. The applicable requirements on processing property right protected data depend on multiple factors including applicable property rights and used license models. In particular, in the context of IaaS, dealing with intellectual property can be complex, especially if corporate customers utilise licensed software [102, part 3 recital 143]. Also, the lawfulness of temporal duplication in clouds is discussed controversially [102, part 3 recital 151], which is particularly relevant for the cloud provider when migrating virtual resources or creating backups. Due to this complexity, a case-by-case analysis of property rights and their impact on outsourcing data processing to the cloud is necessary, which is beyond the scope of this thesis. Therefore, property right protected data and their processing in the cloud are not further investigated in this thesis. Possibilities for applying methods identified in this thesis to processing of property right protected data are disused in the outlook (cf. Section 7.3).

Unprotected data are the remainder of all data that are not considered personal data, business data, or property right protected data. The collection and processing of such data are unregulated and basically legal. Therefore, they can be processed freely and without any additional requirements in the cloud. Before marking data unprotected, the cloud providers should clarify that neither legal norms nor specific legal requirements apply. In particular, it is possible for unprotected data to become personal data, for example by linking unprotected data with a person. In general, unprotected data can be processed any time in the cloud whether safeguards are applied or not. Therefore, they are not further investigated in this thesis.

3.1.2 Basic requirements for data processing in clouds

Lawful cloud computing must comply with applicable legal norms. As a precondition, the cloud customers who are outsourcing data to the cloud must be admissible to collect and process the data.

For personal data, collection and processing is forbidden where the prohibition can be lifted (Art. 7 Data Protection Directive). Generally, the lawfulness of collecting and processing personal data of both the corporate customer's private clients and employees is given by their contractual relationship with the corporate customer (Art. 7 lit. b Data Protection Directive). Alternatively, the data subjects may give their explicit consent (Art. 7 lit. a Data Protection Directive). That can be necessary in the particular case of collecting and processing special categories of personal data, e.g., medical and health data, when the contractual relationship is not sufficient (Art. 8 para. 1 Data Protection Directive). Here, the explicit consent of the data subject is regularly necessary (Art. 8 para. 2 lit. a Data Protection Directive). In order to process personal data without explicit consent, (special) legal permission have to apply.[1] In particular, legal permissions can apply if the necessity of data processing is explicitly given by law (Art. 7 lit. c–f Art. 8 lit. b,c and e second part Data Protection Directive). Further, collection and processing of personal data are only allowed for "specified, explicit and legitimate purposes", i.e., *purpose limitation* (Art. 6 para. 1 lit. b Data Protection Directive),

[1] In general according to Art. 7 lit. c–f Data Protection Directive and for special categories of personal data according to Art. 8 para. 2 lit. b–e Data Protection Directive.

and the processing of personal data must be "adequate, relevant and not excessive in relation to the purposes" (Art. 6 para. 1 lit. c Data Protection Directive). The latter is particularly addressed in German legislation by the *principle of data avoidance* (§3a BDSG). This implies that the nature, scope, and quality of the processing personal data is additionally restricted by the purpose of processing. Generally, the outsourcing contract states whether *purpose limitation* applies and for what purposes the given data are to be processed (Art. 17 para. 3 Data Protection Directive and, in Germany, §11 para. 2 no. 2 BDSG). Therefore, the cloud provider and the corporate customers have to consider applicable purpose limitations when designing the contract. In particular, they have to define adequate measures that have to be implemented in the cloud ensuring the compliance of data processing with *purpose limitation*. Further, "the controller must implement appropriate technical and organizational [sic!] measures to protect personal data" (Art. 17 para. 1 Data Protection Directive).[1] In particular, the "measures shall ensure a level of security appropriate to the risks represented by the processing and the nature of the data to be protected" (Art 17 para. 1 cl. 2 Data Protection Directive). Which measures apply for cloud computing and the necessary safeguards at the cloud provider are investigated in Section 3.3). Another important aspect is that transfer of personal data to third countries generally requires an *adequate level of protection* that is ensured by the third country (Art. 25 para. 1 Data Protection Directive and, in Germany, §4b para. 2 cl. 2 in conj. with §4b para. 3 BDSG). This implies the necessity of *transfer control* for personal data, in particular for recipients located in third countries. Transfer control is investigated in more detail for data transmissions in the context of carrying out data processing in Section 3.2.3, and against the background of the necessity for location-determined data processing in Section 3.5.3.

For business data, collection and processing is generally admissible if it is not prohibited specifically. The nature, scope, and quality of data processing are usually addressed within the context of service descriptions (which are part of the outsourcing contracts) [102, part 2 recital 196 seq.]. Basically, the lawfulness of processing business data within the cloud depends on the lawfulness of outsourcing the related IT processes to the cloud and, further, is clarified by the outsourcing contract between the corporate customer and the cloud provider. The lawfulness of outsourcing an IT process might be regulated by corresponding sectoral legislation. For example, in the German financial sector, the delegation of accountability is banned (§25b para. 2 KWG) [26, part 9 recital 18] and, by German tax law, the accounting generally has to take place inland(§146 para. 2 AO) [26, part 8 recital 9]. More details on sectoral requirements can be found in Section 3.4, which investigates these requirements specifically, including the requirements for and constraints on admissible outsourcing to the cloud. If the outsourcing of the IT process is admissible (or not further regulated) then the lawfulness of data processing in the cloud is regulated by the outsourcing contract.

For outsourcing in general, the contract addresses applicable legal norms and how to deal with multiple jurisdictions [102, part 2 recital 132 seqq.] including place of jurisdiction and applicable legislation [102, part 2 recital 169 seqq.]. Further, an IT-outsourcing contract is

[1] In Germany, technical and organisational measures are addressed by §9 and appendix to §9 cl. 1 BDSG.

generally a *mixed-type contract* in respect to the services delivered[1]. For that reason, the nature, scale, and quality of the outsourced data processing depends on the service description in the contract [102, part 2 recital 196 seq.]. Additionally, rights of use are clarified by the outsourcing contract [26, part 13 ch. B.V.], which are not necessarily limited to property right protected data transferred to the cloud and can also cover rights to the shared results of collaboration [26, part 13 recital 112 seq.] and rights of access to databases stored in the cloud [26, part 13 recital 116]. In conclusion, the cloud provider is obliged to process data in compliance with the outsourcing contract which defines the nature, scale, and quality of data processing.

3.2 Limits and handling of carrying out data processing

To gain a better understanding of the nature and scope of requirements in IT outsourcing, the example of *carrying out data processing*[2] in the German data protection law (§11 BDSG) is investigated. Carrying out data processing is regulated with respect to responsibility of data processing, details of the outsourcing contract, inspection of the processor, lawfulness of transmission, and considering processing of data requiring specific care (e.g., implementation additional safeguards). Therefore, carrying out data processing is an good candidate investigating nature and scope of regulations on IT outsourcing to the cloud.

Carrying out data processing is considered a privilege to lawfully transfer personal data between the *controller*[3] and the *processor*,[4], and therefore it is not considered a transfer to a *third party* (§3 para. 8 cl. 2 BDSG).[5] The privilege of carrying out data processing can only apply if all legal requirements are met.[6] In particular, the assignment of responsibility is not included in carrying out data processing [28, ch. 2 §5 I]. There are also sectoral requirements with an impact on carrying out data processing (cf. Section 3.4). For example, there are additional requirements for the legality of carrying out data processing for personal data in the context of German social security services (§80 SGB X; see also Section 3.4.4). In the context of cloud computing, carrying out data processing basically applies if responsibilities relevant for ensuring that data protection remains with the controller.[7] In the following, carrying out data processing is investigated in the context of processing personal data in the cloud generally, and it is assumed that it is basically applicable in the context of IT outsourcing to the cloud.

The remainder of this section identifies requirements for carrying out data processing with respect to responsibilities of controller and processor, including contract details (cf. Sec-

[1] Contract classification by Nägele and Jacobs [148, ch. II.2]: *Lease*: software and data storage (and IaaS, PaaS, and SaaS cloud services in general [102, part 158 recital 158]); *works contract*: backup and installation, implementation, and modification of software; *service contract*: computational power, maintenance and support of software.

[2] The term refers to "carrying out of processing by way of a processor" (Art. 17 para. 3 Data Protection Directive).

[3] The controller is responsible for the data processing (Art. 2 (d), Data Protection Directive).

[4] The processor performs the data processing (Art. 2 lit. e Data Protection Directive); also mentioned *service organisation* [8].

[5] A third party is an involved person/party "other than the data subject, the controller, the processor" (Art. 2 lit. f Data Protection Directive).

[6] Requirements are defined in §11 BDSG including set-up and execution of carrying out data processing [28, ch. 2 §5 II].

[7] In accordance with the statements of German supervisory authorities [102, part 4 recital 53].

tion 3.2.1), inspection of the processor (cf. Section 3.2.2), lawful data transmission (cf. Section 3.2.3), and dealing with professional secrets (cf. Section 3.2.4). In particular, resulting technical implications for cloud computing and the obligations of cloud providers are identified. To investigate the applicable requirements, European and German legislation are inspected since European legislation establishes the implementation of data protection law by the member states and particularly by Germany.

3.2.1 Processor, controller, and their responsibilities

Basically, the corporate customer contracts the cloud provider to process personal data in the cloud. Therefore, the corporate customer is considered the *controller*. Further, the cloud provider is considered the *processor* since the cloud provider is contracted to act on behalf of the corporate customer.

By being **the controller**, the corporate customer has to ensure legality of the data processing (according to Art. 7 lit. a in association with Art. 6 para. 1 Data Protection Directive and its German implementation by §11 BDSG) including *purpose limitation* (Art. 6 para. 1 lit. b, c Data Protection Directive), quality of the data processing (Art. 6 para. 1 lit. d Data Protection Directive), and compliance with retention and deletion periods (Art. 6 para. 1 lit. e Data Protection Directive; see also Section 3.5.1). Moreover, the controller has to govern the processor generally (Art. 17 para. 3 Data Protection Law) and, by German data protection law, has to inspect the organisational and technical measures of the processor (§11 para. 2 cl. 4 BDSG). Furthermore, the controller is generally responsible for providing information according to the right of access (Art. 12 Data Protection Directive), right to rectify (Art. 12 lit. b Data Protection Directive), and right to object (Art. 14 Data Protection Directive). Additionally, the controller is responsible for notifying the supervisory authority (Art. 18 Data Protection Directive).

Basically, the responsibilities of **the processor** are to "act only on instructions from the controller" (Art. 17 para. 3 Data Protection Directive) and to implement appropriate technical and organisational measures (Art. 17 para. 1 in assertion with Art. 17 para. 3 Data Protection Directive). The instructions from the controller are specified in the contract between controller and processor (§11 para. 2 no. 2 BDSG). Particularly relevant for cloud computing, the contract states necessary technical and organisational measures that have to be implemented by the processor (§11 para. 2 no. 3 BDSG in conj. with §9 BDSG), correction, deletion and blocking of data (§11 para. 2 no. 4 BDSG), the inspection obligations of the processor including internal inspection and inspection by competent supervisory authorities (§11 para. 2 no. 5 in conj. with §11 para. 4 BDSG, particularly regarding §§4g, 10, and 38 BDSG), explicit authorisations on subcontracting (§11 para. 2 no. 6 BDSG), inspection by the controller (§11 para. 2 no. 7 BDSG), processor's obligation to notify on infringements of data protection law and the contract by the controller (§11 para. 2 no. 8 BDSG), and return of data media to the controller and deletion of data stored at the processor at the end of the contract (§11 para. 2 no. 10 BDSG). These clauses will also be included in the upcoming GDPR, and additionally, the processor will be regularly considered the controller if personal data are processed other than as instructed by the controller (Art. 26 para. 4 GDPR) and will have to inform on data breaches (Art. 31 GDPR) [104, ch. 4].

In conclusion, the controller has to require the processor to implement measures (i.e., safeguards) ensuring legal compliance in the cloud and to inspect effectiveness and compliance with safeguards. On the other hand, the processor must follow the instructions of the controller and implement appropriate safeguards. A notable requirement is that the contract explicitly has to state if subcontracting is allowed. In the context of cloud computing, such a clause can be very restricting since cloud providers generally subcontract hardware providers to operate the hardware infrastructure of the cloud. Further investigations concerning the necessary safeguards at the cloud provider for handing subcontracts are located in Section 3.3.4. Another notable requirement is the inspection of legal compliance at the processor. This has to be done by the processor (internally) and by the controller (supervisory). Additionally, inspections by competent supervisory authorities may apply. Due to the complexity of cloud infrastructures (e.g., virtualisation, distributed computing, and multiple involved parties), supervisory inspection can be difficult. Requirements for inspection of the cloud provider and their implementation are investigated in Section 3.2.2 specifically.

3.2.2 Inspection of the processor

The data processing at the processors has to be inspected internally by the processors themselves and must be supervised by the controllers and supervisory authorities (cf. Section 3.2.1). In general, all inspection results must be documented (§11 para. 2 cl. 5 BDSG).

In particular, the **internal inspections by the processor** (i.e., cloud provider) are performed by the processor's data protection officer (§§4f, 4g BDSG)[1] and cover primarily the effectiveness of technical and organisational measures (§9 BDSG) implemented at the processor. Additionally, internal inspections apply to implementing automated searches of personal data (§10 BDSG)

Further, the **supervisory inspection by the controller** (i.e., corporate customer) has to be done before the first time data are processed and on a regular basis during data processing (§11 para. 2 cl. 4 BDSG).[2] The goal of the inspection is to ensure and proof the compliance of the technical and organisational measures with the applicable contract and data protection law. On the one hand this can be done by the controller or by a named investigator acting on behalf of the controller, for example, in form of announced sample checks.[3] On the other hand, it is considered best practice that this should be done by the processor,[4] for example, by proving up-to-date attestation, audit reports done by independent authorities (also in extracts), certification of IT-Security and data protection audits as announced by §9a BDSG. Standards and certifications addressing IT security and data protection law are, for example, the German 'IT-Grundschutz' [31], ISO/IEC 27004 [111], ISO 27005 [109], and the European Privacy Seal for IT Products and IT-Based Services (EuroPriSe) [208].

[1] In this thesis, it is assumed that cloud providers generally are required to commission a data protection officer (cf. Section 3.3.6)

[2] The upcoming GDPR does not regulate these types of inspection [104, ch. 4.b].

[3] This is particularly the case for automated search of personal data (§28 para. 2 cl. 4 in conj. with §10 para. 4 BDSG).

[4] For instance, according to the model contract [82] and the inspection template [83] provided by the Gesellschaft für Datenschutz und Datensicherheit (GDD).

Moreover, the **supervisory inspection by competent supervisory authorities** must be possible generally (§11 para. 2 no. 5 in conj. with §11 para. 4 BDSG and §38 BDSG). This may include providing access to data processing facilities, data processing software and processed data to inspectors of the supervisory authority (§38 para. 4 BDSG).

In conclusion, the cloud provider has to both implement internal inspection and support the supervisory inspection by externals (namely, the corporate customer and supervisory authority). Even if the inspection by the corporate customer is substituted by applying sufficient measures according to an adequate security standard or certification, the inspection by the supervisory authority cannot be avoided generally. This implies that the cloud infrastructure must support external inspection and, particularly, provide the necessary access to all inspected components. This can be complicated in the case of global cloud computing if data are processed in foreign countries where the supervisory authority has no authority (also cf. Section 3.2.3). Additionally, the cloud provider has to document all information relevant to the inspection (including information on the inspection) to guarantee a successful inspection. Moreover, providing access to externals may be a problem if access is granted to data of corporate customers not involved in the inspection (also cf. Section 3.3.5). The ultimate consequence is that the cloud provider has to implement adequate measures for monitoring, documentation, and reporting of compliance if dealing with all the issues mentioned above.

3.2.3 Prohibition and limitation of data transmission

A particular requirement of German data protection law is *transfer control* (According to no. 4 of the appendix to §9 cl. 1 BDSG). While *data transfer* addresses the act of handover, *data transmission* describes the (technical and organisational) implementation of that act. For example, the data transmission into the cloud technically describes the act of handover between corporate customer and cloud provider (usually via a network connection between the IT infrastructure of the corporate customer and the cloud). In the context of carrying out data processing, the transmission between controller and processes is privileged and, therefore, not considered a transfer (§3 para. 4 no. 3 in conj. with §3 para. 8 cl. 3 BDSG; see also Section 3.2.1). However, there are boundaries that have to be considered. In particular, data transmission may imply data transfer to third parties which have not been granted the privilege of carrying out data processing and, therefore, transfer control may apply.

In the remainder of this section, the different scenarios of transmission that can apply in the context of cloud computing are investigated:

Data transmission into the cloud is carried out from the corporate customer to the cloud provider. First of all, it has to be distinguished if the transmission is part of a carrying out data processing or an assignment of responsibilities. For carrying out data processing, the transmission of personal data between controller and processor generally is admissible. However, data transmission into the cloud may result in data transmission to subcontracted third parties (particularly hardware provider and cloud providers) operating computational resources of the cloud infrastructure (which may include cloud services operated by service providers).[1] In such a case, the subcontracting has to be explicitly authorised in the outsourcing contract

[1] In fact, there could be a long chain of subcontracts involving all kinds of legitimate actors (cf. Section 2.2.1).

(cf. Section 3.2.1) or otherwise is prohibited.[1] For assignment of responsibility, transmission of personal data from the corporate customer to the cloud provider requires explicit permission by a legal norm or the explicit consent of all involved data subjects (cf. Section 3.1.2). In such a case, the privilege of carrying out data processing does not apply, and the cloud provider becomes responsible (i.e., controller) for processing personal data in the cloud (cf. Section 3.2.1). Nevertheless, the responsibility for data transmission into the cloud generally lies with the corporate customer since he or she is the controller responsible for processing the transmitted personal data generally (cf. Section 3.2.1). Furthermore, the legality of data transmission to the cloud depends on the recipients involved and their location.[2] Recipients (here: cloud providers and hardware providers) may be located in countries different from the sender's origin (here: corporate customer's origin). In any case, an *adequate level of protection* (Art. 25 para. 1 Data Protection Directive) has to be ensured by the controller (which is generally the corporate customer and for cross-boarder transmissions regularly the cloud provider) at the recipients' location (cf. Section 3.1.2; see also *cross-border transmission* below).

Data transmission within the cloud is carried out by the cloud provider and regularly performed to ensure optimal use of the hardware infrastructure as well as for backup and recovery operations. Such data transmission may result in data transmission to the subcontracted hardware provider. Likewise in the case of data transmission into the cloud, involving subcontractors has to be explicitly allowed within the contract between controller and processor. While the corporate customer is responsible for data processing in the cloud, the cloud provider has to process data as contracted (cf. Section 3.2.1) which may include restrictions on recipients and recipients' location to ensure an *adequate level of protection* (Art. 25 para. 1 Data Protection Directive). In the case of assignment of responsibility, the cloud provider becomes controller for his or her assigned responsibilities including data transmission within the cloud in general. Again (cf. *data transmission into the cloud* above), an *adequate level of protection* has to be ensured by the controller.

Cross-border transmission is the case if sender and recipient are not located within the same country. Generally, there is *choice of law* (According to Art. 6 of Rome I in conj. with Regulation (EC) No 593/2008). Applicable law usually is clarified in the contract, and particularly, it is specified how data protection law and carrying out data processing apply.[3] In the EU/EEA, the lawfulness of cross-border transmissions of personal data to third counties (i.e, to the outside of the EU/EEA) requires an *adequate level of protection* to be ensured in the recipient's country (Art. 25 para. 1 Data Protection Directive) and may require adequate safeguards implemented by the recipients to to ensure protection of personal data (Corresponding to Art. 26 para. 2 Data Protection Directive). A more detailed analysis on admissible transmission to third countries and applicable requirements and restrictions can be found in Section 3.5.3. It is important to note that the privilege of carrying out data processing does not apply to data transmission to third countries since then the recipient is considered a third party §3 para. 8 cl.

[1] For the requirements and responsibilities involved in subcontracting see Section 3.3.4.

[2] In this context, it is significant to distinguish between locations within EU/EEA and third country (Art. 25, 26 Data Protection Directive).

[3] For instance: the *standard contractual clauses for the transfer of personal data to processors established in third countries under Directive 95/46/EC* (URL: http://eur-lex.europa.eu/legal-content/de/TXT/?uri=CELEX:32010D0087; last visited: 30.06.2015).

3 BDSG). In this case, the cloud provider becomes responsible controller for the transmission [86, §4b recital 5]. In any case, the corporate customer remains responsible for the lawfulness of the transmission in general (Art. 6 para. 2 Data Protection Directive in conj. with §4b para. 5 BDSG) and is liable for damage (Art. 23 para. 1 Data Protection Directive in conj. with §7 BDSG). This implies that the corporate customer has a particular interest that the cloud provider complies with transmission restrictions and that compliance with them is a particular topic of inspections (cf. Section 3.2.2)

Transmission to third parties Beside all previously mentioned requirements on transmission, additional requirements may apply when data are transmitted to third parties. If the third party is subcontracted by the cloud provider then this has to be explicitly authorised in the contract between cloud provider and corporate customer (cf. Section 3.2.1). Which requirements particularly apply in this case are discussed in Section 3.3.4. In any case, there has to be explicit permission by a legal norm, which is regularly an *adequate level of protection* at the recipient's location (cf. "cross-boarder transmission" above and Section 3.1.2). Further, non-disclosure agreements and service level agreements may apply additionally if they are stated in the contract between cloud provider and corporate customer.

Automated search of personal data In German data protection law, data transfer using an *automated search of personal data*.[1] is regulated explicitly by §10 BDSG Automated searching is basically allowed for personal data (§10 para. 1 BDSG)[2] and requires additional documentation to specifically support inspections (§10 para. 2 BDSG). Further, explicit authorisation by the responsible supervisory authority is required (§10 para. 3 BDSG). The site requesting the data is responsible for the data transfer (§10 para. 4 cl. 1 BDSG). The site storing the personal data (and making them available for the automated search) can be required to inspect an automated search of personal data and, therefore, has to implement measures for sample checks (§10 para. 4 cl. 2 seqq., BDSG).

In the context of carrying out data processing, regulations on automated searches of personal data may not apply to data transmissions between controller and processor since the controller is generally responsible for implementing legal requirements (§11 para. 1 cl. 1 BDSG). Particularly, §10 BDSG does not apply to the processor (§11 para. 4 BDSG). Outside of the context of carrying out data processing, the cloud provider may be responsible for data transmission (if requesting personal data by automated searching) and additional duties on inspection and documentation may apply (if storing personal data for automated searching).

[1] Corresponding with the definition of Art. 24 para. 2 lit. b Member States' Initiative 2007/C 71/13.
[2] Prerequisites: protection of legitimate interests and proportionality.

3.2.4 Professional secret

Corporate customers may be required to keep professional secrets, e.g., German nursing services[1] and hospitals have obligations of secrecy according to §21 Abs. 1 Nr. 5 SGB IX and §203 StGB.[2] If corporate customers are required to keep professional secrets then a contracted cloud provider and their subcontractors also are required to keep these professional secrets [103, ch. IV.]. Further, cloud providers and their subcontractors might be prosecuted under German criminal code if professional secrets of corporate customers are disclosed. In particular, in case of an offence, they may be considered accomplices and therefore the offender and the accomplice might be treated as equals (§203 para. 3 cl. 2 StGB). Therefore, cloud providers have to be aware if corporate customers transfer data into the cloud which are protected by professional secrets. In such a case, the transfer of the data can be considered a disclosure [102, part 7 recital 65] and therefore illegal without consent of the data subject [172, ch. V] [103, ch. IV] (see also Section 3.4.4). If the transfer is legal, proper safeguards are necessary to protect the professional secret properly. Details on safeguards generally addressing confidentiality and secrecy in the cloud are discussed in Section 3.3.1.

3.3 Necessary safeguards at the cloud provider

Cloud providers are required by law and the contract with the corporate customer to exercise due diligences, for example in the context of carrying out data processing (cf. Section 3.2.1). Therefore, they have to take technical and organisational measures to implement adequate safeguards and ensure an *adequate level of protection* (cf. Section 3.1.2). There exist multiple IT security standards and best practices, providing methods for implementing effective safeguards. For instance, the ISO/IEC 27000-series provides guidelines on implementing information security management (i.e, ISO/IEC 27002 [113] and ISO/IEC 27003 [110] in conj. with ISO/IEC 27001 [112]). The ISO/IEC 27000-series has also been adopted in the German 'IT-Grundschutz' [30] which provides a comprehensive catalogue of best practices [31].

This section will discuss the legal requirements for safeguards at the cloud provider and identify recommendations and best practices on their implementation in IT security standards. A particular focus is on requirements for technical measures in clouds.

3.3.1 Confidentiality

Frequently, processed data are considered confidential. For personal data, this is the case in general since the data secret applies (Art. 16, Data Protection Directive and §5 BDSG). Further, professional secrets may apply to personal data (§203 StGB; see also Section 3.2.4). In addition, the secrecy of personal data may apply in a specific context, e.g., the protection of

[1] IT outsourcing and cloud computing are highly relevant for nursing services due to demands on revenue-cost-optimisation and automated data processing [100]. There already exist cloud offerings specialised for nursing services, e.g., Mobile Pflege Cloud by the Deutsches Medizinrechenzentrum (DMRZ) (`http://www.dmrz.de/mobile-rechtssichere-pflegedokumentation-nach-gesetzlichen-vorgaben.html` (last visited: 30.06.2015).

[2] For the obligation of secrecy of nursing staff, see Roßbruch [172, ch. II.].

personal data in German social law (§35 SGB I and §78 SGB X) and the secrecy of telecommunications in German (§88 TKG and §206 StGB). For business data, outsourcing contracts may state non-disclosure agreements. In particular, the trade secret in Germany (§§17, 18 UWG) and German tax secrecy[1] (§30 AO) may apply for business data.

To ensure confidentiality, technical and organisational measures have to be implemented by the data processing parties. This is particularly in the context of data protection the case (Art. 17 para. 1 in conj. with recital 46 Data Protection Directive and, in Germany, §9 BDSG). For example, in Germany data protection law, technical and organisational measures addressing confidentiality are defined specifically. Corresponding to the appendix to §9 cl. 1 BDSG, control measures have to be implemented for physical access, data access, data usage, and data transfer to ensure that personal data are not disclosed to unauthorised persons. In particular, data encryption is considered an adequate control measure (cl. 3 of appendix to §9 cl. 1 BDSG). This implies that the cloud provider generally has to implement these control measures when processing personal data in the cloud.[2] Another example of requirements for technical and organisational measures addressing confidentiality can be found in the context of processing tax data. In German tax law, access control measures have to be implemented to protect tax data from unauthorised access (recital 103 GoBD). On the other hand, if tax data are encrypted it has to be ensured that there is access in decrypted form (recital 134 GoBD; see also Section 3.3.3). Regularly, service contracts contain SLAs for confidentiality particularly in the form of NDAs. These have to be implemented by the cloud provider as well.

In conclusion, data confidentiality and secrecy also have to be ensured in the cloud. This may require implementing access and transfer control including data encryption. However, these measures must not restrict access by (or transfer to) competent authorities.

3.3.2 Authenticity and integrity

Another frequent requirement for data processing is data integrity. To fulfil the service contract, cloud providers have to ensure the correctness of the data processing. Usually, SLAs ensuring data integrity (e.g., protection against manipulation and ensured data quality) are included in the service contract. Closely related to integrity is authenticity, which particularly addresses securely identified data sources, recipients and users. Additionally, authenticity is a basic requirement for ensuring access, usage, and transmission control and, therefore, for ensuring the confidentiality and secrecy of data (cf. Section 3.3.1). Further, data integrity can be of importance in the context of inspections – e.g., inspection of the processor by the controller (cf. Section 3.2.2) – if it is necessary to prove the correctness of documentation.

Data integrity is explicitly addressed by European data protection law (Art. 6 para. 1 lit. d Data Protection Directive), and German data protection law summarises both authenticity and integrity within the requirement on input control (cl. 2 no. 5 of appendix to §9 cl. 1 BDSG). Further, data integrity is of particular importance for processing tax data. For example

[1] Mentioned in Common Position (EC) No 24/2002 (annex "statement of the council's reasons" VIII. 5.).

[2] This also applies in the context of carrying out data processing. Here, the corporate customer is responsible for their implementation, and the cloud provider is required to implement them by the contract with the corporate customer (cf. Section 3.2.1).

in German tax law, tax data must not modified (§146 para. 4 AO) and its immutability has to be ensured explicitly (recital 110 GoBD).

This implies that in the cloud, too, data integrity has to be protected. The cloud provider may be required (for example, by SLAs in the service contracts) to implement measures that support data integrity. If implementing access, usage or transfer control, the cloud provider has to implement authenticity measures to securely identify involved data sources and persons. This is particularly important when subcontractors are involved (cf. Section 3.3.4). Further, authenticity is a prerequisite for implementing multi-tenancy (cf. 3.3.5).

3.3.3 Availability

The availability of data and services within the cloud is essential. Without availability the cloud provider cannot fulfil the service contract, and the corporate customers can neither access their data nor use the contracted services. Moreover, availability is an important requirement addressed in multiple legal norms. In German data protection law, availability control is an explicitly required measure that has to be implemented (cl. 2 no. 7 of appendix to §9 cl. 1 BDSG). Further, German tax regulations require the availability of tax data and related software and hardware[1] for inspection/investigation by the responsible tax office (recital 118 no. 2 and recital 130 GoBD). Furthermore, there are multiple obligations to store data for a specific period of time, for example, German tax data have to be stored for at least six years (§147 para. 3 AO). Additionally, data availability may be ended by deletion obligations, for example, accounting data in German telecommunications services must not be stored longer than six month (§97 para. 3 TKG). If deletion periods are shorter then retention periods than there is a conflict which is solved in the context of German data protection law by replacing deletion with the blocking of the data (§20 para. 3 and §35 para. 3 BDSG and §84 para. 3 SGB X, respectively). An overview of retention and deletion periods of German legal norms investigated in this thesis can be found in Section 3.5.1 (see also table tab:selectionPeriods).

Usually, the responsibility for data availability is with the corporate customers since they are controllers (for processing personal data; cf. Section 3.2.1) or directly considered responsible by applicable legal norms (e.g., tax law requires taxpayers to ensure data availability for tax inspections). However, the cloud provider may also be responsible, for example, upon becoming the controller in the context of transmitting personal data to third countries (cf. Section 3.2.3). Additionally, the outsourcing contract may contain SLAs on data and service availability which is usually the case since this is an important criterion of service quality. Therefore, the cloud provider has to implement measures to ensure the availability of data and services.

3.3.4 Handling subcontractors

If cloud providers contract hardware providers to operate the hardware infrastructure of the cloud then the hardware providers are considered subcontractors. There may also be other subcontractors involved by the cloud provider, i.e., service provider, software vendor, hardware

[1] Software and hardware access are relevant for direct and indirect access by the inspectors (cf. recital 174 and 175 GoBD).

vendor and other cloud provider (cf. Section 2.2.1). The lawfulness of involving subcontractors is generally clarified in the contract between cloud provider and corporate customer. This is particularly the case for carrying out data processing (cf. 3.2.1). In this context, the subcontracts have to correspond to the contracts with the corporate customers and the authority of the corporate customers has to be considered properly (in particular with respect to inspection) [102, part 4 recital 91]. This includes professional secrets, which have to be explicitly addressed in the subcontracts (cf. Section 3.2.4).

This implies that the cloud provider has to involve subcontractors carefully. Safeguards also have to be implemented at the subcontractor level, and if the subcontractor are located in third countries an *adequate level of protection* generally has to be ensured (Art. 25 para. 1 Data Protection Directive). This is particularly true for cross-boarder transmissions in the context of carrying out data processing (cf. 3.2.3). Also, the cloud provider has to ensure that the inspection of subcontractors is possible with respect to fulfilment of orders and implementation of adequate safeguards (also cf. Section 3.2.2). Further, restrictions on transmission may limit the legitimate recipients among available subcontractors, for example, in the case of cross-boarder transmissions in the context of carrying out data processing (cf. 3.2.3). In particular, transmission control might be necessary when involving subcontractors.

3.3.5 Multi-tenancy and rule of separation

Within the cloud, software and data of different corporate customers are processed on the same hardware infrastructure and, due to virtualisation, frequently on the same hardware. Multi-tenancy means that the software and data of a single corporate customer are accessible and modified only by the corporate customer him- or herself. In particular, the software and data of each corporate customer are isolated from the software and data of all other corporate customers. While multi-tenancy is a technical principle, it is also introduced by law, i.e., in general by transmission and access restrictions. Further, for processing personal data the rule of separation applies (cl. 2 no. 8 of appendix to §9 cl. 1 BDSG). The latter includes the segregation of duties, which is not necessarily addressed by multi-tenancy. Segregation of duties ensures that separate duties and particularly contradicting duties are executed by different persons (or parties). Generally, segregation of duties is implemented by organisational controls, but can also be enforced technically, for example, by access control on data, software and hardware. In the context of cloud computing, examples of segregation of duties are separating backup and data processing (for safety reasons) and distributing the hosting of corporate customers among different hardware providers (to support multi-tenancy and transmission control).

3.3.6 Other obligations

Alongside the safeguards mentioned above, there are also organisational obligations addressing process and risk management. Particularly in the financial sector, additional regulations exist that require the implementation of proper risk management (cf. Section 3.4.1).

Moreover, German data protection law requires the commission of a data protection officer (§11 para. 4 no. 2 in conj. with §4f BDSG) if there are more than nine employees involved in the processing of personal data, which is usually but not necessary the case for cloud providers

(automation allows the operation of clouds with a small staff). With the upcoming GDPR, the requirements for when a data protection officer has to be commissioned have changed. The threshold is increased to 250 employees (Art. 35 para. 1 lit. b GDPR). Further, commissioning is necessary if nature, scope and/or purposes "requires regular and systematic monitoring of data subjects" (Art. 35 para. 1 lit. b GDPR). Both is unlikely to apply to cloud providers which are involved in IT outsourcing. Therefore, it is likely that these cloud providers do not have to commission a data protection officer after the new GDPR has entered into force.

For automated data processing within the Cloud, there exist in data protection law notification obligations (Art. 18 Data Protection Directive) which in Germany apply for all data processing parties (§4d para. 1 BDSG). The notification obligation does not apply if a data protection officer is commissioned (Art. 18 para. 2 cl. 2 second part Data Protective Directive and, in Germany, §4d para. 2 BDSG). or if the data subject has given his or her consent to the data processing (§4d para. 3 BDSG).

Further, prior checking of data processing systems may apply (Art. 20, Data Protection Directive and, in Germany, §4d para. 5 BDSG). In Germany, this is particularly the case for processing special categories of data and for purposes of evaluating the data subject with respect to personality including skills, performance and behaviour (§4d para. 5 cl. 2 BDSG). Exceptions are legal obligations to process the data, consent of the data subject, and the necessity to process the data in order to establish the execution or cessation of contractual obligation. In these cases, the prior checking may be omitted.

There also exist several retention, deletion and documentation obligations which are investigated in more detail in Section 3.5.1. In particular, the results of the inspection of the processor by the controller in the context of carrying out data processing (cf. Section 3.2.2) have to be documented (§11 para. 2 cl. 5 BDSG). Also, data processing of financial and tax data are subject of documentation (cf. Section 3.4.1 and Section 3.4.2). Documentation is also helpful in supporting the controller to satisfy the rights of private clients and employees of the corporate customer, for instance, the *right of access* (Art. 12 Data Protection Directive), the *right to rectify* (Art. 12 lit. b Data Protection Directive), and the *right to object* (Art. 14 Data Protection Directive).

If there are cross-border transmissions to third countries by the cloud provider then the cloud provider is obliged to satisfy these rights, too, because of becoming controller for these transmissions (cf. Cross-border Transmission).

All obligations mentioned above are organisational but have implications for the technical implementation of clouds. The cloud infrastructure have to support (or at least not hinder) the implementation of the mentioned organisational obligations. This includes particularly the documentation of the cloud infrastructure and operation for the purpose of inspection by the data protection officer, prior checking, and satisfying retention and documentation obligations.

3.4 Dealing with sectoral requirements

In Section 3.2, the requirements of data protection law were exemplified by carrying out data processing in Germany. European and specifically German data protection law defines high standards for an *adequate level of protection* when processing personal data. Beside data pro-

tection law, there exist other legal norms regulating data processing, outsourcing, and data transfer. This section investigates selected sectoral regulations, which apply in the context of IT outsourcing regularly. In Section 3.4.1, the requirements of the financial sector are exemplified by German legislation. Section 3.4.2 examines German tax law. Export control within the EU/EEA and Germany is investigated in Section 3.4.3. Specific requirements applying in the German medical and healthcare sector are discussed in Section 3.4.4. In the end, outsourcing by public authorities is investigated using the example of the German public sector.

3.4.1 Financial sector

In German legislation, outsourcing of financial services is specifically regulated. For example, there exist regulations for the outsourcing of banking (§§25a, 25b KWG), payment services (§20 ZAG), stockbroking (§§25a, 25b KWG in conj. with §33 para. 2 cl. 1 WpHG), insurance (§64a VAG), and investment management (§80 KAGB). While the regulations on outsourcing banking, payment services and stockbroking are similar (§20 Zahlungsdiensteaufsichtsgesetz (ZAG) corresponds with §25b KWG while §25a KWG specifies additional requirements for stockbroking), the regulations on outsourcing insurance and investment management deviate. In the following, the particular requirements for IT outsourcing to the cloud in these areas are investigated.

Banking, payment services, and stockbroking
The outsourcing of banking, payment services, and stockbroking is in principle admissible but requires proper measures (§25a para. 1 cl. 1 KWG and §20 para. 1 cl. 1 ZAG, respectively). In particular, correctness and regularity of business, services, and business organisation has to be ensured (§25a para. 1 cl. 2 KWG and §20 para. 1 cl. 2 ZAG, respectively), with a special care for segregation of duties (§25a para. 1 cl. 2 no. 3 lit. a KWG; see also Section 3.3.5) and an appropriate and effective risk management (§25a para. 1 cl. 3 KWG and §20 para. 1 cl. 3 ZAG, respectively). Proper measures must include at least:

- identification, evaluation, management, monitoring, and reporting of risks (§25a para. 1 cl. 3 KWG and §20 para. 1 cl. 3 ZAG, respectively);
- appropriate personnel (i.e., experts) and effective technical and organisational implementations (§25a para. 1 cl. 3 no. 4 KWG);
- implementation of an emergency concept, in particular for IT systems (§25a para. 1 cl. 3 no. 5 KWG);
- periodic inspection for appropriateness and effectiveness (§25a para. 1, cl. 5 KWG); and
- complete documentation (§25a para. 1 cl. 6 no. 2 KWG).

Further, the responsibilities of the management are not permitted to be delegated (§25b para. 2 cl. 4 KWG and §20 para. 1 cl. 4 ZAG, respectively). The outsourcing corporate customer remains fully responsible (§25b para. 2 cl. 5 KWG and §20 para. 1 cl. 5 ZAG, respectively).

For stockbroking, the outsourcing must not change the legal relationship of the corporate customer to his or her customers or responsibilities (§33 para. 2 cl. 2 WpHG), nor must it change the terms of the operating permit (§33 para. 3 cl. 1 WpHG). Moreover, outsourcing shall not hamper the supervisory authority and other examiners from inspection and controlling, especially not, if cross-border transmissions to the outside of Germany or the EEA exist (§25b para. 3 KWG and §20 para. 1 cl. 6, 7 ZAG, respectively). The contract between the corporate customer and the cloud provider has to be in written form and must cover the privileges of the corporate customer, including authority and cancellation, and the responsibilities of the cloud provider (§25b para. 3 cl. 3 KWG and §20 para. 1 cl. 8 ZAG, respectively). Subcontracting is possible and has to address these requirements within the subcontracts of the cloud provider [102, part 8b recital 69 seq.].

Additionally, there is a regulation called Mindestanforderungen an das Risikomanagement (MaRisk) [32] which is applicable according to §25a para. 6 KWG and specifies the requirements for risk management and its implementation. These requirements apply if outsourcing is significant (MaRisk AT 9 in conj. with §25a para. 2 KWG) [102, part 8b recital 16] and, in particular, to cloud computing if services and functions of a financial institution are administered [102, part 8b recital 19 seq.]. The latter is regularly the case if the outsourcing of services and functions is based on IaaS [102, part 8b recital 28]. If outsourcing is significant it is decided on a case-by-case basis [102, part 8b recital 32]. Alongside with formal requirements for the outsourcing contract – which are similar to those of carrying out data processing [102, part 8b recital 43] (see also Section 3.2.1) – there exist requirements on service quality [102, part 8b recital 44 seqq.] including: (1) specifications on the place of fulfilment, (2) implementation of monitoring and control mechanisms,[1] (3) service availability plus responsibility in case of malperformance, (4) monitoring, inspection and controlling by the corporate customer, authority of the corporate customer, and (5) subcontracting which requires explicit approval by the corporate customer [102, part 8b recital 70]. Further, there are requirements for access control and multitenancy [102, part 8b recital 71 seqq.] (see also Section 3.3.1 and Section 3.3.5). In particular, safeguards ensuring integrity, availability, authenticity, and confidentiality have to be implemented according to existing IT security standards [102, part 8b recital 78 seqq.], like the ISO/IEC 27000-series [112] [113] [110] [111] [109] and the German IT-Grundschutz [30] [31]. Furthermore, business continuity has to be considered by including incident response and recovery measures [102, part 8b recital 82 seq.].

Insurance Outsourcing in the context of insurance is regulated in Germany by the Versicherungsaufsichtsgesetz (VAG). According to §64a para. 4 VAG, the requirements for insurance agencies are similar to those for financial institutes [102, part 8b recital 92]. In particular, the outsourcing has to be properly executed, and it must neither impair the management and monitoring capability of the management nor hamper the supervisory authority from carrying out inspection and controlling (§64a para. 4 cl. 1 VAG). Further, the necessary authority to obtain information and issue instructions has to be implemented within the outsourcing contract (§64a para. 4 cl. 2 VAG). This implies that cloud providers are bound by instruction from

[1] To mitigate the loss of control, when outsourcing to the cloud, it is proposed that control mechanisms should also cover the virtualisation management [102, part 8b recital 46].

the corporate customers, similar to outsourcing of banking, payment services and stockbroking (and carrying out data processing, too).

Investment management Outsourcing in the context of investment management is specifically regulated in German legislation by §36 Kapitalanlagegesetzbuch (KAGB). Outsourcing is allowed under particular restrictions, one of which is that the service organisation has sufficient resources and an experienced and reliable management (§36 para. 1 no. 2 KAGB). A further restriction is that outsourcing to third countries requires ensured cooperation between the German federal agency and the supervisory authority of the third country (§36 para. 1 no. 4 KAGB). Additionally, continuous inspection of the service organisation is required (§36 para. 1 no. 8 KAGB). According to §36 para. 36 para. 6 KAGB, subcontracting is allowed if the corporate customer has explicitly approved it, the German federal agency is notified, and the requirements for outsourcing stated §36 para. 1 no. 2–8 KAGB have been implemented.

Conclusion In the financial sector, outsourcing to the cloud has requirements that are similar to those for carrying out data processing (cf. Section 3.2) apply. The cloud provider is bound by instructions from the corporate customers and has to implement safeguards covering confidentiality, integrity, authenticity, and availability. In particular, access control and multi-tenancy have to be ensured. Further, safeguards have to be ensured by implementing IT security standards like the ISO/IEC 27000-series. Additionally, service execution has to be monitored including inspections and reporting. This particular includes documentation. Additionally, there exist restrictions on cross-board transmissions and access by competent supervisory authorities has to be ensured if necessary.

3.4.2 Tax data in the cloud

If business processes are outsources to the cloud this can have relevance in the context of taxation. For example, if using IaaS then the cloud may be considered a permanent establishment which is then relevant for the taxation of corporate customers.[1] Another important aspect in the context of taxation is the inspection by competent supervisory authorities. Particularly in the case of electronic accounting, the high degree of regulation in Germany is higher than in other countries where often no regulations exists [102, part 6 recital 78]. Therefore, German tax law is a good example to investigate those requirements that can apply to outsourcing of *tax data* (i.e., data relevant for taxation) to the cloud.

In Germany, keeping accounts (according to §238 HGB) is generally regulated by the AO. In the case of electronic accounting (which is regularly the case when outsourcing tax data to the cloud), the German regulation, Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff (GoBD)[2] [33], provides additional and more specific requirements on electronic

[1] Generally, it can be assumed that this is only the case for private clouds and depends case-by-case [102, part 6 recital 26 seqq.]. On the other hand, for cloud provider leasing and buying server is regularly relevant [102, idid.].

[2] Applicable since 1st Jan. 2015; replacing Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen (GDPdU) and Grundsätze ordnungsgemäßiger DV-gestützter Buchführungssysteme (GoBS).

accounting. In general, electronic accounting is admissible but requires integrity of archived data [102, part 6 recital 77]. In particular, immutability during transmissions has to be ensured [102, part 6 recital 90]. Further, documentation and internal controlling is required as well as changing of existing entries must not possible [102, part 6 recital 77]. An particular interesting observation is that beside accounts[1] also correspondence via email is considered tax data [102, part 6 recital 84]. This implies that tax regulations not only apply to cloud resources directly connected to accounting (e.g., virtual machines hosting accounting software and tax data achieves) but also to other cloud resources like email servers and archives. This obliges the corporate customer to verify if tax regulations apply to data outsourced to the cloud and clearly specify how the cloud provider has to process them. Otherwise, the corporate customer may violate German tax law if the cloud provider, for example, transfers data to third countries. Generally, tax data have to be stored inland but a storage outside is possible according to §146 para. 2 cl. 1 AO and §14b para. 2 UStG but the competent revenue authority has to be informed on location of data processing and involved cloud providers and hardware providers [102, part 6 recital 87] (see also Section 3.5.3).

The GoBD also addresses external audits and inspections by competent revenue authority which have access privileges according to §147 para. 6 AO (recital 158 GoBD). In this context, access has to be provided to all IT systems containing tax data in general including documentation of the IT systems and an overview of all data stored within (recital 159 GoBD). If tax data are outsourced to the cloud also IT systems of the cloud may be included in audits and inspections. This can lead to multiple problems, for example, if the prosecution demands physical access to cloud resources [102, part 7 recital 10 seqq.]. Particular issues are the location of cloud resources in foreign countries and storage of tax data with data of other corporate customers on the same hardware resource. Section 3.5.2 generally investigates confiscation and distraint in the cloud. More detailed information on how the cloud provider has to deal with such cases can be found there. Even the usual case of an inspection by the competent revenue authority bear complex issues. An example is the *direct access (Z1)*[2] where data have to be accessed via the original IT system (i.e., the cloud infrastructure and the associated remote system) only but remote enquiry is inadmissible (recital 165, GoBD). Remote enquiry is a basic concept of cloud computing. Therefore, the requirements of direct access are technically unaccomplishable for clouds. A possible solution might be to migrated the cloud resources to a hardware resource which is physical accessible by the competent revenue authority. In this case, the access may be direct but possibly lacks of a remote software which is required to read out and visualise the data. Consequently, how to deal best (and without causing any legal or technical issues) with direct access in clouds remains legally and technically unanswered. An alternative for inspecting cloud resources might be *indirect access (Z2)* (recital 166 GoBD). In this case, the corporate customer (or commissioned third party) analyses the tax data as instructed by the competent revenue authority and provides read-only access on the analysed data. Also possible is the *data medium provision (Z3)* (recital 167 GoBD). Here, all tax data

[1] A comprehensive overview of documents relevant for record retention is provided by Hilbert [102, part 6 recital 81 seqq.].

[2] GoBD distinguishes by three types of access: direct access (Z1), indirect access (Z2), and data medium provision (Z3) (recital 165–167 GoBD).

are provided on a data medium differently from hardware resources used in the cloud, and particularly, no access to the IT system is necessary.[1]

In conclusion, outsourcing of tax data to the cloud is generally admissible but requires the authorisation of the competent revenue authority. The corporate customer has to ensure that the cloud provider processes tax data in compliance with applicable requirements. These requirements particularly cover providing information on the location of the data-processing system and of the cloud provider (plus any subcontractors involved), ensuring the integrity of archived data, and providing unhampered, timely and automated access to tax data for external audits and inspections by competent revenue authorities.

3.4.3 Export control and dual-use

Generally, exportation is regulated on an international, European, and national level [102, part 8f recital 5]. A particular focus of export control is on *dual-use* technology, i.e., technology which can be used for both civil and military purposes[102, part 8f recital 7]. In the EU/EEA, export control with respect to dual use is regulated by the Dual-Use Regulation (EU/428/2009). According to Appendix I of this regulation, software and technology for telecommunication (category 5 part 1 Dual-Use Regulation) and information security (category 5 part 2 Dual-Use Regulation) may be affected. In particular, restrictions on cryptographic software from other regulations – including implemented national law of European member states – must be followed (category 5 part 2 Dual-Use Regulation). In Germany, export control is regulated by the Außenwirtschaftsgesetz (AWG) and the Außenwirtschaftsverordnung (AWV) with a particular focus on military technology which is specified in the export control list.[2] [102, part 8f recital 15]

In the context of cloud computing, export control applies if software and technology mentioned by the export control list is exported [102, part 8f recital 34]. This is particularly the case if data (technology or software mentioned by the export control list) are transmitted (even temporarily) to third countries and if cloud services are offered to the corporate customer located in third countries (on screen visualisation in third countries is sufficient) [102, Idid.]In this context, every cross-boarder transmission can be relevant for export control including transmissions (1) to servers located abroad [102, part 8f recital 40], (2) between member states of the EU/EEA (since export control is also regulated on national level) [102, part 8f recital 43 seq.], (3) to clouds located in third countries, (4) via third countries (for each involved country) [102, part 8f recital 47 seqq.], and to authorised employees of corporate customers temporarily located abroad [102, part 8f recital 51 seq.]. This implies that for software and technology mentioned by the export control list any cross-boarder transmission (into, within or from the cloud) is subject to export control and requires transmission control (similar to carrying out data processing; cf. Section 3.2.3).

In general, the corporate customer is responsible for transmitting data relevant to export control into the cloud, and for ensuring compliance with export control. Further, it is not clarified when the cloud provider can also become responsible, in particular, if the cloud provider

[1] In this context access is rather inadmissible (recital 167 cl. 2 GoBD).

[2] Latest version: attachment 1 to the AWV as amended on 05.08.2013 (BGBl. I no. 45, pp. 2898 seqq.).

knew about restrictions of export control or should have known about them [102, part 8f recital 76]. In any case, providing cloud services to corporate customers abroad can be inadmissible according to §34 AWG if the corporate customers are established in countries listed for embargo[1] or if the corporate customers or their representatives are listed for embargo in the context of terrorism.[2] Therefore, it is recommended to clarify within the outsourcing contract that export control applies and that data with relevance to export control are not allowed to be stored in foreign countries [102, part 8f recital 77].

In conclusion, transmission control has to be implemented in clouds if data with relevance to export control (i.e., software and technology mentioned on European and national export control lists) are processed. Transmission control has to cover (1) hardware resources utilised for hosting cloud resources, (2) routes and links used for data transmissions, and (3) client systems accessing the cloud externally. Further, the corporate customer is responsible for informing the cloud provider fo whether export control applies and to which countries transmission is allowed. Additionally, providing cloud services to corporate customers abroad can be inadmissible if there is an embargo of the countries where the corporate customers are established or if the corporate customers or their representatives are listed for embargo. Where there is an embargo, the cloud provider has to ensure the enforcement of export control.

3.4.4 Medical and healthcare sector

In the medical and healthcare sector, additional requirements – along with general requirements on processing personal data (cf. Section 3.1.2 – may apply to the processing of personal data. Within the EU/EEA, heath data are considered a *special category of data* (according to Art. 8 para. 1 Data Protection Directive) and their processing is forbidden where the prohibition may be lifted (according to Art. 8 para. 2 Data Protection Directive). The cases for lifting of the prohibition are similar to those for processing of personal data generally (cf. Section 3.1.2) but with increased thresholds. For example, it is possible that "laws of the member [state] provides [sic!] that the prohibition [...] may not be lifted by the data subject's giving his [or her] consent" (Art. 8 para. 2 lit. a Data Protection Directive).[3] Another example is, that lawful processing of employee data requires adequate safeguards (Art. 8 para. 2 lit. b Data Protection Directive). Further, additional obligations of secrecy may apply. For example in Germany, if the corporate customer is processing personal data in the context of social security services (according to SGB X) then the obligation of secrecy applies according to the service contracts with the service providers (§21 para. 1 no. 5 SGB IX). Additionally, the corporate customer

[1] German embargo list dated 15.06.2015 provided by the Bundesamt für Wirtschaft und Ausfuhrkontrolle (BAFA) (on the Internet: `http://www.ausfuhrkontrolle.info/ausfuhrkontrolle/de/embargos/uebersicht/uebersicht_laender_bezogene_embargos.pdf` last visited: 30.06.2015).

[2] In Germany, the BAFA provides information on terrorism related embargos (on the Internet: `http://www.ausfuhrkontrolle.info/ausfuhrkontrolle/de/embargos/terrorismus/index.html` last visited: 30.06.2015) and provides the Handbuch der deutschen Exportkontrolle (HADDEX) including embargo lists in digital form (on the Internet: `http://www.bundesanzeiger-verlag.de/de/aw-portal/exportkontrolle/produkte/sanktionslisten.html` last visited: 30.06.2015).

[3] For instance, in German, any transmission of patient data for the purpose of billing (according to SGB V) by hospital facilities and penal doctors to private service companies is inadmissible (Bundessozialgericht (BSG), Decision dated 10.12.2008 – B 6 KA 37/ 07 R, head note 2).

may be obliged to a professional secret (cf. Section 3.2.4) which is usually the case for medical facilities and healthcare service providers, for instances in Germany, according to §203 StGB [172, ch. II]). In this case, the data subject usually must agree to the revelation of the secret information to the cloud provider and involved subcontractors.[1]

If health data are transmitted to the outside of the EU/EEA then particular care is required. If the recipient's location does not have an *adequate level of protection* the recipient must be able to ensure the necessary safeguards [86, §4b recital 7]. When evaluating the appropriateness of the safeguards, it should be considered that the risk for the data subject is assumed to be high [86, §4b recital 11]. Furthermore, the recipient must be informed of the given purpose of the transmission and told that he or she must follow the *purpose limitation* [86, §4b recital 19].

Moreover, all requirements that apply to personal data also apply to health data. For instance, in Germany, this includes obligations to (1) implement technical and organisational measures (§9 BDSG in conj. with appendix to §9 cl. 1 BDSG), (2) cooperate – particularly on inspections – with competent supervisory authorities (§38 para. 3–4 BDSG), and (3) allow for the inalienable rights of the data subjects (§6 para. 1 BDSG) including the right to information (§§19 and 34 BDSG).

In conclusion, the outsourcing of the processing of health data to the cloud is admissible but usually requires the implementation of safeguards ensuring the confidentiality and secrecy of the data (see also Section 3.3.1). In particular, the implementation of transmission control is paramount since cross-boarder transmissions may be prohibited if the recipient cannot ensure adequate safeguards (see also 3.2.3). Further, inspection of competent supervisory authorities has to be supported by the cloud provider which implies that adequate documentation and reporting mechanisms are required (see also Section 3.2.2).

3.4.5 Public sector

Specifically for the public sector, restrictions on outsourcing sovereign duties and responsibilities as well as requirement for tendering procedures are of particular relevance.

In general, the **outsourcing of sovereign duties and responsibilities** is restricted by national law, for instance in Germany, according to Art. 33 para. 4 Grundgesetz (GG) [137]. Outsourcing of sovereign duties and responsibilities to the private sector might be prohibited but, for example in Germany, technical support can be outsourced if it does not include the authority to make decisions on sovereign duties and responsibilities [98, Ch. 5 recital 102 seqq.].

Further, **regulations on tendering procedures** may apply. The framework for regulations in the context of tendering procedures is the Government Procurement Agreement (GPA) of the World Trade Organization (WTO) in the year 1996 which is implemented in European and German law [102, part 8c recital 8]. In the context of cloud computing, the European Directive 2004/18/EG is most relevant, and in Germany, the forth part of Gesetz gegen Wettbewerbsbeschränkungen (GWB) and Section 2 of Vergabe- und Vertragsordnung für Leistungen Teil

[1] In Germany, the data subject's consent is considered sufficient for such revelations [172, ch. V] [103, ch. IV].

A (VOL/A) apply [102, part 8c recital 9–10].[1] Relevant for cloud providers is that they may have to provide information on subcontracting (Art. 25 Directive 2004/18/EG corresponding to Art. 71 para. 2 Directive 2014/24/EU) including the nature and involvement of subcontractors. Further, "obligations relating to employment protection provisions and the working conditions which are in force in the place where the works are to be carried out or the service is to be provided" have to be taken into account (Art. 27 para. 2 in conj. with Art. 27 para. 1 Directive 2004/18/EG.[2]) This includes employment protection provisions and the working conditions of the premises of subcontractors such as hardware providers. Consequently, the cloud provider has to know which hardware providers will be involved and the consequences on employment protection provisions and the working conditions. For example, if 24/7 support is required and night work has significantly increased costs then the cloud provider may decide to locate the support at multiple locations in different time zones to avoid night work. Another example are cloud services covering tasks that are executed by employees (like the *amazon machanical turk*).[3] In this case, employees may have to be selected according to applicable requirements of employment protection provisions and working conditions. This implies that the place of fulfilment can be relevant, and therefore, fulfilment in specific countries can be inadmissible. In such cases, the cloud provider has to select subcontractors not established in inadmissible countries. Another issue is that, in the particular context of German data protection law, it is controversially discussed whether it is legal when German authorities involve processors from outside of the EU/EEA [137, ch. III.c.I]. For legal certainty, subcontracting processors from outside of the EU/EEA should be avoided when processing personal data for German authorities.

To avoid these issues on outsourcing within the public sector, governmental clouds (i.e., operated under governmental authority) have been established in multiple member states of the European Union [69]. Within ENISA's report on governmental clouds of 2013 [69, ch. 5], recommendations on existing solutions were given, including the definition of a uniform governmental cloud strategy and further (1) implementation of standard procedures on monitoring, data handling, and data migration, (2) implementation of "a regulatory framework to address the locality problem", (3) a guarantee of compliance with European and national law, (4) implementation of a common framework for SLAs, and (5) enhancement of security measures in governmental clouds including "research on governmental cloud security" and "privacy enforcement". This implies that governmental clouds require technically implemented safeguards particularly ensuring security of processed data and compliance with applicable legislation, which is in conformance with the observations made on necessary safeguards at the cloud provider in Section 3.3.

[1] Directive 2004/18/EG will be replaced by Directive 2014/24/EU on 18.04.2016. The following investigations are based on the current Directive 2004/18/EG. Difference from Directive 2014/24/EU are mentioned where applicable.

[2] Corresponding to Art. 71 para. 6 in conj. with Art. 18. para. 2 Directive 2014/24/EU

[3] Amazon mechanical turk, on the Internet: https://www.mturk.com/mturk/welcome (last visited: 30.06.2015).

3.5 Special requirements

Within European and German legislation, there are requirements that are of particular interest for cloud computing since they are applicable generally and have technical implications for the data processing and the implementation of cloud computing. This section discusses the requirements for retention, deletion and documentation using the example of German legal norms investigated in the previous sections. Further, the specific case is inspected where cloud data are subject of search and confiscation. In this context, it is particularly relevant how cloud providers should cooperate with investigative authorities and grant access to stored data. Moreover, requirements for the location of data processing and cross-boarder transmissions are analysed. In particular, the need to determine the location of IT systems processing data in the cloud is investigated using the examples of European and particularly German data protection law as well as German tax law.

3.5.1 Retention, deletion and documentation

When processing data requirements to store data for a given time period or delete data within a given time period can apply. In this section, time periods and requirements for storing and deleting data that are relevant for cloud computing are exemplified by German legislation.

Requirements for storing data for a given time period, i.e., the *retention period*, apply if data have to be archived for documentation and inspection reasons. For instance according to German telecommunication law, the origin and recipients of personal data have to be stored for two years (§34 para. 1a BDSG). Within *retention periods*, requirements to keep data available and to grant access (e.g., for inspections by competent supervisory authorities) can apply, which are addressed in the following *provision obligations*. For example, *provision obligations* regularly apply to German tax data (§147 para. 5 and 6 AO). Further, *retention periods* and *provision obligations* can be coupled with documentation requirements, e.g., documentation of the IT systems utilised for processing tax data (according to recital 159 GoBD).

Requirements to delete data within a given time period, i.e., *deletion period* apply if storing data is no longer necessary or even prohibited. Deletion periods regularly apply in the context of German data protection law, when processing personal data. Here, personal data have to be deleted if their storing is not longer admissible or necessary for the purposes for which they were originally collected and processed (§35 para. 2 and §20 para. 2 BDSG). Particularly for commercial data processing, the necessity has to be controlled within particular time periods (§35 para. 2 no. 4 BDSG).

In general, storage and deletion requirements apply to the controller, but they regularly also apply to contracted processors and their subcontractors if the controller has to ensure that his or her contracted processors and their subcontractors comply with these requirements. Therefore, requirements regarding retention, provision, deletion and documentation and how to deal with them are usually implemented within outsourcing contracts and subcontracts.

Table 3.1 provides an overview of *retention periods* and *deletion periods* for data categories addressed by German legal norms discussed in this thesis. Generally, for all data categories *deletion periods* apply, except for tax data. Traffic data have to be deleted immediately after termination of connection (§96 para. 1 TKG). Another observation is that there are long time

Table 3.1: Selection of retention and deletion periods in German legislation

Data category	Retention period	Deletion period	Regulation(s)
Processing of personal data			
– personal data	—	for no longer than is admissible or necessary for the purposes[a][b]	§35 para. 2 and §20 para. 2 BDSG
– personal data in the context of social security services	—		§84 para. 2 SGB X
– for purposes of address trading and advertising (origin and recipients of transmitted data)	2 years		§34 para. 1a BDSG
– patient data	10 years[c]		§10 para. 3 Berufsordnungen der Ärztekammern
Tele-media services[d]			
– personal data (in respect to service usage)	—	immediately after termination of usage[a]	§13 para. 4 no. 2 TMG
– usage data	—	for no longer than is necessary for accounting[a]	§15 para. 4 TMG
Telecommunication services[e]			
– customer data	—	after contract termination at the end of the following calendar year	§95 para. 3 TKG
– accounting data	—	for not longer then 6 month after billing	§97 para. 3 TKG
– traffic data (generally)	—	immediately after termination of connection	§96 para. 1 TKG
– data relevant for security authorities	—	after contract termination at the end of the following year	§111 para. 4 TKG
Keeping of accounts			
– tax data with relevance to provide evidence	6 years	—	§147 para. 3 AO; §§238 and 257, HGB; recital 113 seqq. GoBD
– tax data with relevance to cadastre and accounting	10 years	—	
– procedure documentation (generally)	10 years	—	
– procedure documentation (only relevant with relevance to provide evidence)	6 years	—	

[a] If there are contradictory *retention periods* then blocking instead of deletion applies (§35 para. 3 and §20 para. 3 BDSG).
[b] For commercial data processing, check of necessity three years after saving and four years after execution of affairs (§35 para. 2 no. 4 BDSG)
[c] Even longer in exceptional cases, e.g., according to radiation protection law and radiation control law up to 30 years.
[d] According to §1 para. 1 TMG (usually information society services according to Directive 2000/31/EC)
[e] According to §3 no. 24 TKG

periods of retention (up to ten years and in the medical context even longer). Particularly for tax data, secure storage and availability for inspection by competent authorities have to be ensured for multiple years. If data are no longer needed before the *retention period* expires blocking access to data can be substituted for its deletion, both generally (§20 para. 3 and §35 para. 3 BDSG) and in the context of social security services (§84 para. 3 SGB X). In the case of IT outsourcing to clouds, blocking has to be supported by the cloud provider. Further, the corporate customer has to require the cloud provider to ensure blocking access to data and deletion of data and inspect compliant implementation by the cloud provider (cf. Section 3.2.1). Furthermore, cloud providers have to ensure that their subcontractors implement the requirements for blocking access and deletion of data compliantly, too.

In conclusion, the corporate customers may be required to store, provide or delete data within given time periods. When processing data to which such requirements apply in the cloud, the cloud provider has to be able to ensure proper storage and deletion of the data. This

includes ensuring data availability for access and blocking data access if necessary. Further, subcontractors of the cloud provider have to ensure this as well. Otherwise, there is a risk of violating *retention periods* and *deletion periods* and data processing in the cloud may not be lawful. Additionally, the corporate customer may be required to document data processing and utilised IT systems. In such a case, the cloud provider has to be able to provide documentation on data processing within the cloud and necessary documentation on IT systems. Without the support of the cloud provider on documentation, the corporate customer may not be able to comply with documentation requirements and data processing in the cloud may not be lawful. This is particularly the case for processing tax data within the cloud (cf. Section 3.4.2).

3.5.2 Search and confiscation in the cloud

If a corporate customer is subjected to prosecution then data stored in the name of that corporate customer within the cloud can be affected by a warrant to search or confiscate. In such a case, the investigative authorities are regularly granted the power not only to search the corporate customer's premises but also the cloud infrastructure storing data of that corporate customer. For instance, in Germany, access to the corporate customer's data stored in the cloud by German authorities is generally legitimated according to §94 Strafprozeßordnung (StPo) [102, part 7 recital 31 seq.]. If the requested data are not provided voluntarily the investigative authority can confiscate the data (§94 para. 2 StPo). For cloud providers, this is relevant when one of their corporate customers is searched (in Germany: according to §§102 seqq. StPo) and the prosecution is entitled by court to confiscate data (according to §98 StPo) [102, part 7 recital 34]. In such a case, it is recommended that the cloud provider be able to provide the data, which are subjected to search and potentially confiscation freely[1] and without infringement of requirements by civil law [102, idid.]. Further, the cloud provider may be required to perform a so called *'quick freeze'*, i.e., "to preserve and maintain the integrity of that computer data for a period of time as long as necessary, up to a maximum of ninety days, to enable the competent authorities to seek its disclosure" (Art. 16 para. 2 Convention on Cybercrime (CETS no. 185)), which is not explicitly implemented in German legislation but applicable within the limits of §§94,95 StPo [102, part 7 recital 35]. Generally, confiscation potentially covers all data accessible within the searched premises (in Germany: §110 para. 3 StPo)[2] [102, part 7 recital 35 seqq.]. This includes searching data stored in the cloud using corporate customers' IT systems for access [102, part 7 recital 37]. Further, the search of servers abroad can be complicated since this usually requires the cooperation of the country's authorities at the searched server's location [102, part 7 recital 40].[3]

[1] Moreover, the cloud provider is required to handover the data on demand (§95 para. 1 StPo).

[2] The application of §110 para. 3 StPo on data stored in clouds is arguable, since on the one hand legitimacy for searching is only given if here is an urgent risk of losing this data otherwise but on the other hand cloud providers regularly offer services protecting against data loss (e.g., backup) [102, part 7 recital 39 seqq.].

[3] When searching servers remotely (i.e., online searching), it can happen that the search is unintentionally expanded to foreign countries, if the cloud servers are distributed globally and the location of accessed servers is not visible to the investigators. On the other hand, it is unlikely (but not impossible) that the authorities in the foreign county will take notice of such an online search.

Hilbert [102, part 7 recital 50 seqq.] provides recommendations for how cloud providers (and corporate customers) should prepare for being searched to ensure cooperation with the investigative authority and avoid infringements of civil law. First of all, the assignment of a responsible contact person is recommended, and the contact person should be entitled with all necessary competence and discretionary power to ensure internal cooperation and handing over of requested data [102, part 7 recital 51 seqq.]. Further, technical and organisational measures are recommended including logically separated storages for each corporate customer and the ability to reproduce stored data on a physically separated data medium [102, part 7 recital 53]. In particular, before deleting any data the investigative authority should be contacted [102, part 7 recital 58]. This implies that (if technically possible) automated deletion should be paused during search. Blocking is possibly an adequate replacement during search (see also Section 3.5.1). Additionally, it can be necessary to revoke access privileges of corporate customers being subjected to prosecution [102, part 7 recital 95].

In conclusion, the cloud provider should be able to legally support authorities on search and confiscation in a way that unrelated data and services remain unaffected. This includes ensuring the rules of separation and the confidentiality, integrity and availability of unrelated data. Further, measures for isolating data and for limiting access of authorities to isolated data are necessary. These measures are similar to the measures that are necessary for supporting inspections by supervisory authorities and other authorised examiners, e.g., likewise in the financial sector (cf. Section 3.4.1) or according to tax law (cf. Section 3.4.2).

3.5.3 Necessity for location-determined data processing

In global clouds (cf. Remark 2.5), hardware resources are located in multiple countries, and therefore, data transmissions within the cloud possibly constitute cross-boarder transmissions (see also Section 3.2.3). Cross-boarder transmissions can have a severe impact on whether data processing is legally compliant. For example, data processing within certain countries may require additional security precautions and/or specific prerequisites, e.g., if processing personal data in third countries not having an *adequate level of protection* (cf. Section 3.2.3), and can even be inadmissible, e.g., if processing German tax data abroad without permission of the competent revenue authority (cf. Section 3.4.2). There exist multiple legal norms in Europe and Germany addressing requirements for admissible locations for data transfer, data processing and outsourcing. This section investigates specifically location-related requirements in legislation and their technical implications for cloud computing using the example of European and particularly German data protection law as well as the example of German tax law.

3.5.3.1 Location constraints in European and German data protection law

In European data protection law, data transfer to third countries generally requires an *adequate level of protection* to be ensured at the recipient's location (Art. 25 para. 1 Data Protection Directive). Whether or not the ensured level of protection is adequate "shall be assessed in the light of all the circumstances surrounding a data transfer operation" including "the nature of the data, the purpose and duration of the proposed processing operation [...], the country of origin and country of final destination, the rules of law [...] in force [...] and the professional

rules and security measures which are complied with in that country" (Art. 25 para. 2 Data Protection Directive). If "a third country does not ensure an *adequate level of protection*", "measures necessary to prevent any transfer of data of the same type to the third country in question" shall be taken (Art. 25 para. 4 Data Protection Directive). Further, there exist derogations when transfer of data to third countries not ensuring an *adequate level of protection* are admissible (Art. 26 Data Protection Directive) including the data subject's consent (para. 1 lit. a idid.), necessity for a contractual relationship with the data subjects (para. 1 lit. b, c idid.) or protecting their vital interests (para. 1 lit. e idid.), necessity and legal obligation in the context of important public interest and legal claims (para. 1 lit. d idid.), explicit authorisation by the member state if "the controller adduces adequate safeguards" (para. 2 idid.).

In Germany, Art. 25 and 26 of the Data Protection Directive are implemented by §§4b and 4c BDSG, respectively. The necessity for an *adequate level of protection* is implied by the requirement to protect the data subject's legitimate interests to object to a data processing in countries that does not have an *adequate level of protection* (§4b para. 2 cl. 2). Derogations in terms of Art. 26 Data Protection Directive are implemented identically by §4c BDSG. In particular, §4c para. 2 cl. 1 specifies that explicit authorisation can given by a competent supervisory authority provided that adequate safeguards are ensured.

Consequently, it is of particular importance for the admissibility of data transfer in European and German legislation whether an *adequate level of protection* is ensured at the recipient's location or adequate safeguards are ensured by the recipient. Here, the question arises as to which countries have an *adequate level of protection* and what are adequate safeguards if not.

The EU commission has approved *adequate levels of protection* for "Andorra, Argentina, Canada (commercial organisations), Faroe Islands, Guernsey, Israel, Isle of Man, Jersey, New Zealand, Switzerland, Uruguay and the US Department of Commerce's Safe Harbour Privacy Principles" [65].[1] For all other countries there are additional precautions necessary to ensure the implementation of adequate safeguards. The European commission provides two sets of standard contractual clauses that are recommended to be included within the contract between sender and recipient (Decision 2001/497/EC and Decision 2004/915/EC). The first set of standard contract clauses "for the transfer of personal data to third countries which do not ensure an *adequate level of protection*" in the annex of Decision 2001/497/EC particularly covers obligations of the data importer, i.e., recipient (Clause 5 idid.) including:

- to enure that there is no reason to believe that legislation applicable to the data importer prevents the fulfilment of the contract (Clause 5 lit. a idid.);
- "to process the personal data in accordance with the mandatory data protection principles" (Clause 5 lit. b idid.);[2]

[1] It is of particular interest that there are doubts on the effectiveness of Safe Harbor Privacy Principles in practice [102, part 4 recital 238] and the use of standard contractual clauses have been suspended in Germany [102, part 4 recital 242]. Even if agreements on ensuring adequate safeguards exist, the transmission may be prohibited by competent supervisory authorities [102, part 4 recital 241].

[2] There are two options on mandatory data protection principles: that of appendix 2 and that of appendix 3 of the Decision 2001/497/EC. Both are investigated subsequently to the standard contract clauses (along with those of Annex A Directive 2004/915/EC).

- "to deal promptly and properly with all reasonable inquiries from the data exporter or the data subject [...] and cooperate with the competent supervisory authority" (Clause 5 lit. c idid.); and

- "to submit its data processing facilities for audit" (Clause 5 lit. d idid.).

The governing law is the law of the member state in which the data exporter is established (Clause 10 idid.). The second set of standard contract clauses "for the transfer of personal data from the Community to third countries (controller to controller transfer)" provided in the annex of Decision 2004/915/EC specifies additional obligations of the data importer (II. idid.) including:

- "appropriate technical and organisational measures to protect the personal data [...], and which provide a level of security appropriate to the risk represented by the processing and the nature of the data to be protected" (II. lit. a idid.) including that the access to the personal data "respect[s] and maintain[s] the confidentiality and security of the personal data" (II. lit. b idid.);

- personal data will be processed for specified purposes (II. lit. d idid.)

- personal data is not provided "to a third party data controller located outside the European Economic Area (EEA) unless it notifies the data exporter about the transfer" and adequate protection is provided by the third country, the third party data controller signs an approved data transfer agreement, data subjects have the opportunity to object, and, if applicable, data subjects have given unambiguous consent for onward transfers of sensitive data (II lit. i idid.).

Again, the governing law is the law of the member state in which the data exporter is established (IV. idid.) with optional exceptions (II. lit. h idid.). Along with the standard contract clauses, mandatory data protection principles apply covering (i) *purpose limitation*, (ii) rights of access, rectification, erasure and blocking of data, and (iii) restrictions on onward transfers.[1] Additionally, appendix 3 Decision 2001/497/EC and annex A Decision 2004/915/EC address (1) data quality and proportionality, (2) transparency,[2] (3) security and confidentiality, (4) special categories of data/sensitive data, (5) direct marketing/data used for marketing purposes, and (6) automated decisions. In summary, the mandatory data protection principles cover basically all data protection principles of European data protection law, which is the purpose of the standard contract clauses.

In conclusion, the corporate customer and the cloud provider have to be aware of data transfers to third countries, which are regularly implied by cross-boarder transmission to recipients established outside of the EU/EEA. In such a case, the country at the recipient's location has

[1] These requirements are covered by all versions of mandatory data protection principles, i.e., those of appendix 2 and appendix 3 Decision 2001/497/EC and of annex A Decision 2004/915/EC.

[2] Transparency is particularly regulated by the upcoming GDPR where "transparent and easily accessible policies with regard to the processing of personal data and for the exercise of data subjects' rights" become necessary (Art. 11 para. 1 idid.) and requires the differentiation of involved parties' obligations as well as the protection of the legitimate interests of cloud providers [104, ch. 3.c].

to ensure an *adequate level of protection* or the recipient has to provide adequate safeguards.[1] In the context of data transmission within the cloud, this implies the necessity of transmission control based on the location of subcontracted cloud and hardware providers as well as on the origin of corporate customers (due to being controllers for outsourcing data processing to the cloud). Moreover, adequate safeguards particularly include restrictions on onward transfers which generally are implemented by technical and organisational measures on transmission control. Therefore, transmission control ensures location-determined data processing is paramount for the processing of personal data in global clouds.

3.5.3.2 Location constraints in German tax law

In Germany, account books (according to §238 HGB) generally have to be kept and stored within Germany (§146 para. 2 cl. 1 AO). As an exception, the keeping and storage of electronic accounts outside of Germany may be authorised by the competent revenue authority[2] upon written application (§146 para. 2a cl. 1 AO) and under restriction of explicit permission and the following preconditions (§146 para. 2a cl. 2AO):

- the location of the data-processing system, as well as name and address of the recipient(s) (i.e., cloud provider plus involved subcontractors) is known;
- the taxpayer (i.e., corporate customer) complies with her or his duties according to §§90, 93, 97, 140–147, and 200 para. 1 and 2 AO;
- access to data for the purpose of external audits/inspections by the competent revenue authority is granted; and
- taxation is not impeded.

Moreover, the retransfer of the data to German territory must be possible at any time (§146 para. 2a cl. 3 AO) as well as access, including that data "can be rendered readable without undue delay and can be processed automatically" (§147 para. 2 cl. 1 no. 2 AO).

This implies that the cloud provider has to be able to inform the corporate customer on all possible locations of data processing and all involved recipients beforehand of outsourcing tax data to the cloud. Further, the cloud provider has to ensure that during the outsourcing tax data is processed only at communicated locations and by communicated recipients. It is possible that communicated locations and recipients do not necessarily cover all possible locations and recipients, which then implies the necessity of implementing transmission controls by location and recipient within the cloud. This is particularly the case if the cloud provider subcontracts cloud/hardware providers after establishing the outsourcing. Additionally, the cloud provider has to ensure the availability of tax data for the purposes of external inspections and retransfer

[1] Exceptions may apply, for instance, to the data subject's consent, but are not necessarily applicable for all data transmission of a single corporate customer. Therefore, case-by-case specifications by the corporate customer and case-by-case decisions by the cloud provider would be necessary for every data transmission, which is not regularly practical for automated data processing. Instead, a basis for decision-making is required, which is universally valid. Such a basis can be the presence of an *adequate level of protection* or adequate safeguards

[2] For example, customs authority for customs and competent tax authority for profit tax [102, part 6 recital 108].

to German territory. In particular, access has to be granted without undue delay. Consequently, high availability constraints apply when processing tax data within the cloud. This further limits possible recipients for transfering tax data to since cloud/hardware providers do not necessarily provide the same level of availability due to location and hardware inhomogeneity (cf. Def. 2.2 and Remark 2.7, respectively). Moreover, applicable requirements on legitimate recipients regularly vary with the corporate customer, since approval is given on an individual basis. Consequently, it is necessary for the cloud provider to decide carefully (in advance and during processing) and for every single corporate customer, which cloud and hardware providers are involved in the processing of outsourced tax data.

3.5.3.3 Conclusions on location constraints on cloud computing

Regulations on admissible locations for data processing are addressed by multiple European and German legal norms. Besides data protection law and tax law – which both exemplify requirements on location-determined data processing – there also exist other examples. For instance, there exist export controls on military related technology, which regulates admissible recipients and recipients' countries (cf. Section 3.4.3). But also outside of the European Union there exist regulations on admissible locations for data processing. The NIST identifies the issue of "data location" and takes the National Archives and Records Administration (NARA) regulations on storing federal documents in the United States of America (USA) as an example. Consequently, the recipients of data transmissions and their location are highly relevant for the admissibility of data transmissions. It is necessary to implement transmission control.

In the context of cloud computing, the recipients of data transmissions are regularly subcontracted cloud and hardware providers since they operate the IT systems where the data are processed. Generally, cloud providers have to be aware of data transmissions' recipients, target locations and if data transmissions are cross-boarder. However, it is usually in the knowledge of the corporate customer whether restrictions on data transmission apply or not. Consequently, it is necessary for the corporate customer to communicate applicable restrictions to the cloud provider. Further, the cloud provider has to be able to consider these restrictions when assigning cloud resources for processing data and transmitting data within the cloud, particularly when involving subcontracted cloud and hardware providers. This includes the selection of subcontractors by *adequate level of protection* and ensured safeguards.

In conclusion, when implementing transmission control in the cloud, the cloud provider has to consider (1) the legal framework conditions (including an *adequate level of protection*) at the recipient's location, (2) the safeguards ensured by the recipient (including implemented security measures), (3) the nature of the data transmitted, and (4) the origin of the data and the corporate customer.

3.6 Conclusions on technical requirements

There are multiple legal norms at a European and German level addressing technical requirements which are also applicable to IT outsourcing to the cloud. First of all, the data protection law regulating requirements for processing personal data, but also sectoral requirements like

tax law and export control may apply when processing data in the cloud. The technical implications of these regulations are not necessarily the same but often similar technical requirements can be found in multiple legal norms resulting in necessary safeguards that have to be implemented by the cloud provider (cf. Section 3.3). Many legal norms address requirements on confidentiality, integrity and availability. Also, there exist multiple regulations with respect to storing data including retention and deletion requirements (cf. Section 3.5.1). Moreover, transmission control is a technical requirement which is regularly necessary when processing personal data and tax data (cf. Section 3.5.3). It also applies when export control is compulsory (cf. Section 3.4.3). Another important aspect is the supervisory inspection of cloud providers. This includes controlling cloud providers by corporate customers as well as formal inspections by competent supervisory authorities including inspections with respect to data protection (cf. Section 3.2.2) and tax inspections (cf. Section 3.4.2). In the context of the criminal code, search and confiscation by investigative authorities can also occur (cf. Section 3.5.2). In all these cases (i.e., inspection, search and confiscation), the cloud provider has to be able to provide access to requested data and IT systems. Further, cloud providers are required to ensure multi-tenancy and the rule of separation (cf. Section 3.3.5) which require access control mechanisms.

Whether technical requirements apply and to what extend they have to be implemented depends on which legal norms are applicable and also on the contractual agreements between cloud providers and corporate customers. This section concludes the technical requirements which derive from the legal requirements identified in this chapter previously. Of particular importance is the identification of necessary safeguards and security measures that have to be implemented by the cloud providers, which are described in Section 3.6.1. Further, Section3.6.2 reflects the necessity to formulate applicable safeguards and security measures in technically enforceable security policies. The safeguards and security measures which have to be implemented and enforced in clouds are concluded in Section 3.6.3. In Section 3.6.4 the implications for compliance management are summarised, and Section 3.6.5 finally sums up all conclusions on technical requirements in clouds.

3.6.1 Identification of the necessary level of security

For cloud providers, it is necessary to ensure an adequate level of security when processing corporate customers' data. What level of security is adequate depends on the applicable requirements of legal norms corresponding to data categories processed and of contractual agreements between cloud provider and corporate customer. When transferring personal data to third countries, there has to be an *adequate level of protection* ensured at the target location (cf. Section 3.5.3.1). Otherwise, processing of personal data at the target location generally is inadmissible and may be admissible only in specific cases including restrictions of additional safeguards by the cloud provider. Also the processing of other categories of data require safeguards with respect to implemented security measures. For example, availability and integrity constraints apply when processing tax data (cf. Section 3.4.2), and the processing is usually restricted to specified countries (cf. Section 3.5.3.2). Generally, corresponding legal framework conditions and compliance with these has to be considered when processing data within the cloud. The corresponding legal framework conditions depend on the categories of processed data and their origin as well as the location where corporate customer, cloud provider and in-

volved subcontractors are established. Consequently, the cloud provider has to select carefully the hosting locations by legal framework conditions (corresponding to processed data and to hosting location) and the level of security due to implemented security measures and effective safeguards.

Both the legal framework conditions and level of security are classified by the necessary, ensured and effective set of conditions and measures, which are defined as follows.

Definition 3.1 (Legal framework conditions) *Generally,* legal framework conditions *describe a set of conditions (i.e., safeguards and obligations) within corresponding legal frameworks. More specifically, the* necessary legal framework conditions *are the set of conditions according to corresponding legal frameworks that has to be ensured by legal frameworks at locations of data processing, i.e., at the establishment of the controller or processor (for instance, the* adequate level of protection *defines legal framework conditions, which have to be ensured when processing personal data in third countries). Further, the* ensured legal framework conditions *specify the set of conditions ensured by the legal framework at a specific location of data processing (for instance, safeguards ensured by German data protection law). Furthermore, the* effective legal framework conditions *define the set of conditions for corresponding legal frameworks that apply when processing a specific set of data.*

Definition 3.2 (Level of security) *Generally,* level of security *describes a set of security measures. More specifically, the* necessary level of security *is the set of security measures that has to be implemented according to* effective legal framework conditions *and contractual agreements. Further, the* ensured level of security *specifies the set of security measures required by* ensured legal framework conditions *and, additionally, applicable contractual agreements, i.e., at the establishment of the controller or processor. Furthermore, the* effective level of security *defines the set of security measures implemented at a specific location of data processing.*

Before data processing, it is necessary to validate if the *ensured legal framework conditions* satisfy the *necessary legal framework conditions*.[1] The result of the validation of the *ensured legal framework conditions* can be three different cases:

1. The *ensured legal framework conditions* are satisfied. Then, it is usually allowed to process the data and the *necessary level of security* is specified according to the *ensured legal framework conditions*.

2. The *ensured legal framework conditions* are not satisfying but can be healed by applying additional safeguards. In this case, the *necessary level of security* is specified by the *necessary legal framework conditions*.

3. The *ensured legal framework conditions* are not satisfying and cannot be healed by applying additional safeguards. In this case, the data processing is generally inadmissible.

[1] This is generally the case if there is no cross-boarder transfer, for example, when processing personal data only in Germany. Further, in terms of European data protection law, this means: when processing personal data in third countries, there has to be an *adequate level of protection*.

To validate the legal framework conditions (and therefore identify the *necessary level of security*) the following pieces of information are paramount:

- the **location** of data processing including storage, backup and support/administration (e.g., Germany, EU/EEA, third country);
- the **category of data** (data type) that is processed (e.g., personal data, business data, or tax data);
- the **origin** of the processor and controller of the data processing as well as of the data subject; and
- the **applicable requirements** from contracts and service level agreements between the cloud provider and the cloud customer.

This implies that that the cloud provider has to be able to identify the *necessary level of security* using the information mentioned above and to allocate cloud resources accordingly. To this end, it is important to make the required information technically available within the cloud. Furthermore, an information model is needed to describe the applicable rules, covering location, data type, and origin as well as the applicable requirements from contracts and service level agreements. The information model can then be used to communicate the requirements between corporate customer, cloud provider and subcontractors and to enforce the *necessary level of security* including location-determined data processing (cf. Section 3.5.3).

3.6.2 Security policies

The implementation and enforcement of the *necessary level of security* requires the definition of security policies in a technically enforceable form. It must be possible to express all applicable security constraints and requirements which includes the mapping of data categories with safeguards and measures that have to be ensured. To support multi-tenancy, the rules have to be expressible and distinguishable by each cloud customer's preferences and requirements. Moreover, the rules of both the cloud customers and the cloud provider have to be expressible. Also, it should be possible to identify and cope with conflicts between different sets of rules that may apply at the same time. To support compliance reviews and forensic analysis, it is important that the security policies be legible and comprehensible.

This is in compliance with the cloud security guidelines of the CSA providing best practices on content and implementation of security policies [46, Sec. 7.7.1] and the ISO 27001 standard providing general guidelines on implementation of security policies [112].

3.6.3 Implementation and enforcement of safeguards

The requirements specified within the security policies have to be implemented and enforced by the cloud provider. To this end, the cloud provider has to provide safeguards including effective security measures implemented in the cloud. These safeguards aim to protect the data processing on behalf of the cloud customer and support agreed service quality and legal compliance.

3.6.3.1 Basic security measures

Basic security measures aim to protect the confidentiality, integrity, and availability of data generally. They also provide the basis for ensuring the authenticity of the data and involved entities as well as the non-repudiation of service usage, access control, secure processing of data and transmission control. Additionally, basic measures cover data security including backup and recovery mechanisms as well as methods for aggregation and anonymisation of personal data. Also secure deletion and blocking of data (if deletion is not allowed due to *retention periods*) are included. Further, they ensure the technical enforcement of the separation of duties and the segregation of data and their processing. In classic outsourcing scenarios, these requirements are well understood and best practices of IT security standards provide detailed guidelines on their implementation (e.g., ISO 27005 [113] and the German 'IT-Grundschutz Kataloge' [31]).

3.6.3.2 Access control

To protect entrusted data, the cloud provider has to ensure that data are accessed and processed only by authorised entities. This includes the distinguishing between corporate customers, subcontracted cloud and hardware providers and the cloud provider, and has to support different privileges of employees and the entitled entities of these bodies. Therefore, the cloud provider has to implement an identity management system and access control mechanisms, for example role-based access control. Further, all these mechanisms have to support logging and documentation if it is necessary to retrace data access for inspections or legal procedures. In addition, access control mechanisms have to support legal access by competent supervisory and investigative authorities, limiting the granted access to a necessary minimum.

3.6.3.3 Transmission control

Transmission control technically ensures that data transmissions are only performed if they are admissible. To this end, it has to ensured that the recipient is authorised to process the data and that the *ensured level of security* and *ensured legal framework conditions* at the recipient's location is adequate.[1] Therefore, each data transmission as well as allocation and migration of virtual resources must comply with applicable transmission restrictions (which can vary for different corporate customers). The decision is made on the recipient's *ensured level of security* and enforces the *necessary level of security*. This means it is necessary to identify the recipient, the recipient's location, the transmitted data category, its origin, and the requirements originating from contracts and service level agreements (cf. Section 3.6.1). Further, transmission control mechanisms have to support logging and documentation in case it is necessary to retrace data transmissions for inspections or legal procedures.

3.6.3.4 Countermeasures and incident response

The cloud management has to be able to deal with disturbances and irregular events like errors and attacks. These may be caused by the cloud provider him- or herself, a cloud customer

[1] In the context of European data protection law, this implies an *adequate level of protection*.

or an external entity. Therefore, it is necessary to implement mechanisms to monitor, prevent and terminate infringements and malicious behaviour including authorised incident response teams. Countermeasures and incident responses should be monitored and documented to provide evidence for the compliance and proportionality of any taken action. Best practices on countermeasures and incident response are provided by IT security standards like the German 'IT-Grundschutz' [31] and other security guidelines like the ones provided by the CSA [46].

3.6.4 Monitoring, documentation, and reporting of compliance

To achieve legal compliance, the *necessary level of security* has to be implemented by effective safeguards and security measures. Here, compliance monitoring is an effective method to ensure this by measuring and documenting the effectiveness and implementation of safeguards and security measures. In particular, the logging and reviewing of data processing and storage, attempted and granted access, and administrator activities is necessary. Compliance reporting to corporate customers is of particular importance for IT outsourcing to the cloud. It empowers the corporate customer to monitor the cloud provider's compliance and can support prescribed inspections by corporate customers.

An important aspect is the protection against misuse and manipulation. Only authorised entities must have access to monitoring and reporting mechanisms. Further, documentation (including logging data) and reports have to be integrity protected to ensure their evidence. Also monitoring, documentation and reporting itself have to be documented and reviewed to ensure proportionality and protection against misuse.

3.6.5 Final conclusion

For legal compliance, the cloud provider has to implement and enforce several safeguards that support the cloud customer's requirements for security and legal compliance. Therefore, the cloud customer needs to be able to communicate these requirements to the cloud provider and in turn the cloud provider needs to be able to report on the compliant implementation of these requirements. In particular, it is important to deal with the requirements of each customer individually, since the requirements of each customer may be different. Within the cloud infrastructure, the cloud management process needs to assign virtual resources based on the physical location of the hardware resource and its *ensured level of security*. As a result of the legal analysis, this leads to the technical requirement of location determined data processing, which enables decision making and enforcement based on an *ensured level of security* and the *necessary level of security*. How the technical requirements can be addressed in clouds is investigated in Chapter 4.

Chapter 4

Technical analysis of cloud computing and supporting legal compliance

Having identified the technical requirements which derive from legal norms applicable to IT outsourcing to clouds, it is possible to investigate the technical capabilities of clouds to address these requirements (Objective 2). Based on these requirements, it is possible to investigate the technical implications of cloud infrastructures when it comes to their compliance with the legal requirements the legal requirements identified in Section 3.6 (Objective 3). This is done with a focus on the utilisation, provisioning, and hosting of virtual resources in IaaS cloud infrastructures for IT outsourcing .

To achieve this, it is necessary to understand the technical characteristics of cloud infrastructures first. Particularly, knowledge of the operation of virtual and hardware resources and their management in the cloud is required. Based on that knowledge it is possible to understand how security measures and safeguards can be implemented to achieve legal compliance. Finally, the technical capabilities of compliance monitoring and reporting are investigated.

The terminology and structure of cloud environments is investigated in Section 4.1. Therefore, virtual resources, hardware resources and the cloud management process are described based on the observation of existing cloud infrastructures. The result is an entity-relationship model which is formulated in an ontology's notation and serves as a basis for all further investigations and descriptions of cloud infrastructures and their behaviour. In Section 4.2, the abilities of cloud security management to satisfy the identified legal requirements are analysis based on literature and existing implementations. In particular, the academic void and the shortcomings of the current practice are identified enabling the classification of challenges in the domain of security management addressed in this thesis. In Section 4.3, the technical capacity to support compliance management in the cloud is analysed. Again, the academic void and shortcomings of current practice are identified, which forms a basis for classifying the challenges in the domain of compliance management addressed in this thesis.

4.1 Towards an IaaS cloud computing ontology

To understand the technical capacity to implement legally compliant data processing in clouds, it is necessary to understand the structure and operation of cloud infrastructures and how they provide cloud services technically.

In this section, a cloud computing taxonomy for IaaS is specified. For this taxonomy, an analysis of existing cloud infrastructures is performed including standards and best practices in data centre design and management as well as cloud computing reference architectures. As a result an entity-relationship model is defined that provide a comprehensive description of virtual resources, hardware resources and the cloud management process. The model is formulated using the formal notation of an ontology and, thereby, is the first step towards an ontology on IaaS cloud infrastructures. This has the advantage that the model is ready for formal verification in respect of its plausibility and being self-contained. Such formal verifications are outside the scope of this thesis and are not further investigated. However, using the systematic methods of ontology construction supports the characteristics of plausibility and being self-contained. How the provided model can be verified in respect of these characteristics is discussed in the outlook (cf. Section 7.3).

In general, cloud infrastructures can be classified (1) by infrastructure elements required for managing cloud services, security, and privacy, and (2) by infrastructure elements required for service orchestration (cf. NIST's cloud computing reference architecture [134]). The management elements provide functions for the overall control and operation of the cloud infrastructure including the interaction with the cloud customer and cloud provider. The elements for service orchestration are organised in a layered structure covering 1) the cloud services (i.e., virtual resources for IaaS), 2) the abstraction and control of computing resources, and 3) the hardware resources.

Figure 4.1: Infrastructure of an IaaS cloud provider according to NIST reference architecture [134].

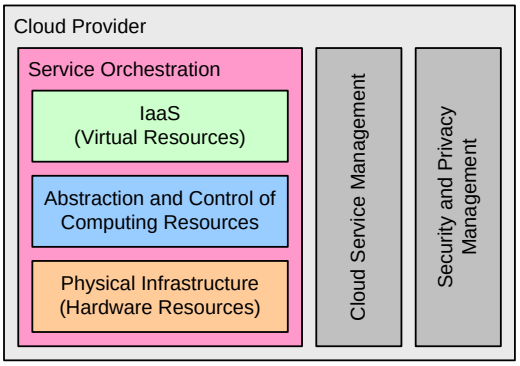


Figure 4.1 provides an overview of the cloud infrastructure elements of an IaaS cloud provider. Beside the service orchestration, there are management elements for provisioning the cloud services to the customer (including the management front-ends for cloud customers and provider as well as accounting) and for ensuring the secure operation of the cloud infrastructure and protecting the cloud customer's privacy.

In the following, virtual resources, hardware resources, and the cloud management process in IaaS cloud infrastructures are specified in an entity-relationship model. First, the formal model of description is given (cf. Section 4.1.1). Then, the cloud customer's view of virtual resources (cf. Section 4.1.2) and the hosting site's view of hardware resources (cf. Section 4.1.3) are described by type and relevant properties. Finally, the cloud provider's view of the cloud infrastructure and the cloud management process are specified (cf. Section 4.1.4).

4.1.1 The entity-relationship model (using an ontology's notation)

For the purpose of defining the cloud entities and their relationships, a mathematical model is used defining classes and relations of the objects described in the notation of an ontology as follows.

Definition 4.1 (Object [Ontology]) *An object $\overline{o}$ is considered an item of interest classified by the ontology and has a defined set of object properties describing the object. The notation of the set of object properties is defined as follows. $\overline{o}|P := \{\overline{p}_1, ..., \overline{p}_n\}$ with $n \in \mathbb{N}$. An object can be a **composition** of a defined set of objects, i.e., $\overline{o} := \{\overline{o}_1, ..., \overline{o}_m\}$ with $m \in \mathbb{N}$.*

An example of an object is a virtual machine instance with the properties of having hardware support and being highly available. Further, a virtual machine instance can be considered a composition of, for example, a virtual CPU, virtual memory and virtual storage.

Definition 4.2 (Class [Ontology]) *A class $\mathbb{C}$ is a set of objects having a defined set of properties in common, where:*

$$\mathbb{C} := \{\overline{c}_1, ..., \overline{c}_n\} \text{ with } n \in \mathbb{N} \text{ and } \overline{c}_i \text{ is an object for } i \in \{1, ..., n\} .$$

*Each Object that is element of $\mathbb{C}$ is also considered an **instance** of $\mathbb{C}$.*

Definition 4.3 (Extends-relation) *The* extends-*relation $\sim_{ex}$ is defined as a relation between two classes $\mathbb{A}$ and $\mathbb{B}$ as follows.*

$$\mathbb{A} \sim_{ex} \mathbb{B} :\Leftrightarrow \mathbb{A} \subset \mathbb{B} .$$

*Class $\mathbb{A}$ is considered **specialisation** of class $\mathbb{B}$.*

Definition 4.4 (Is-associated-relation) *The* is-associated-*relation $\sim_{ass}$ is defined as a relation between two classes $\mathbb{A}$ and $\mathbb{B}$ as follows.*

$$\begin{aligned} &\mathbb{A} \sim_{ass} \mathbb{B} :\Leftrightarrow \\ &\forall \overline{b} \in \mathbb{B} \exists \overline{a} \in \mathbb{A} : \overline{a} \in \overline{b} \text{ and} \\ &\quad \forall \overline{a} \in \mathbb{A} \exists \overline{b} \in \mathbb{B} : \overline{a} \in \overline{b} . \end{aligned}$$

*Class $\mathbb{A}$ is considered **component** of class $\mathbb{B}$.*

Definition 4.5 (Is-property-relation) *The* is-property-*relation* $\sim_{isp}$ *is defined as a relation between two classes* $\mathbb{A}$ *and* $\mathbb{B}$ *as follows.*

$$\mathbb{A} \sim_{isp} \mathbb{B} :\Leftrightarrow$$
$$\forall \overline{b} \in \mathbb{B} : \overline{b}|P = \{\overline{p}_1, ..., \overline{p}_n\} \Rightarrow \exists i \in \{1, ..., n\} : \overline{p}_i \in \mathbb{A} \text{ (with } n \in \mathbb{N}\text{) and}$$
$$\forall \overline{p} \in \mathbb{A} \exists \overline{b} \in \mathbb{B} : \overline{b} \in \overline{b}|P \,.$$

Class $\mathbb{A}$ *is considered* ***property*** *of class* $\mathbb{B}$*.*

4.1.2 Classification of virtual resources

In IaaS, virtual resources can be classified as computational resources (i.e., virtual machines), data storage (i.e., virtual storage), or communication (i.e., virtual links connecting end systems and virtual network services like Quality of Service (QoS) management) [227]. Therefore, virtual machines, virtual storage, virtual links, and virtual network services are good candidate classes for classifying computing resources in IaaS (i.e., virtual resources). Virtual resources are highly relevant for investigating legal compliance in clouds, since the data of corporate customers are processed technically inside of virtual resources. For corporate customers using an IaaS cloud for IT outsourcing, virtual resources are the type of cloud resources they interact with.

4.1.2.1 Reference cloud infrastructures

There exist comprehensive surveys looking at existing cloud infrastructures, which can be used to identify cloud infrastructures that are representative of IaaS. To the best of the author's knowledge, the most recent[1] surveys including commercial cloud infrastructures were performed by Rimal et al. [170] and Prodan and Ostermann [165] in 2009, and Zhang et al. [228] in 2010. Each of the surveys investigates 14 cloud infrastructures (23 different ones in total) and uses a taxonomy – which conforms to the cloud computing paradigm introduced in Section 2.1 – for classifying and comparing the cloud infrastructures. The survey performed by Rimal et al. focuses on the underlying technology of cloud infrastructures and does not cover a classification on computing resources. Prodan and Ostermann distinguish in their survey computing resources by CPU, memory, storage, and hosted operating system. While the analysis of CPU-related resources is multi-dimensional (e.g., cores and architecture), the other categories are investigated in a single dimension (i.e., size for storage and memory, and architecture of hosted operating system). Communication resources are not covered by the survey. In the work of Zhang et al., the implementations of Amazon EC2 (which is part of AWS), Windows Azure, and Google App Engine are compared. The comparison summarises technical details on data centre architectures, distributed file systems, and distributed resource management. All three surveys show that IaaS cloud infrastructures are very similar and usually differ in technical details like resource capacities or underlying hypervisor technology. All three surveys are dated in 2009 and 2010, and therefore, they can be considered outdated with respect to provided information on performance and technology, and with respect to completeness of hosted

[1] As of June 2015

services. However, the conceptual observations are still up to date and can be validated easily by inspecting current cloud infrastructures.

Further, there exist several comprehensive surveys on open(-source) cloud infrastructures. To the best of the author's knowledge, the most recent[1] surveys on open cloud infrastructure were performed by Endo et al. [64] in 2010, and Mahjoub et al. [136] and Voras et al. [215] in 2011. Endo et al. investigate in their survey seven open cloud infrastructures. They provide insight into the computational architecture and resource management of the infrastructures and compare them. The survey by Mahjoub et al. covers six different open cloud infrastructures and focuses on hypervisor technology. Additionally, it provides information on the infrastructure and provided virtual resources. Voras et al. investigate in their survey nine different open cloud infrastructures with respect to storage, virtualisation, management, security, and vendor support. As a conclusion, Red Hat Cloud Foundations, OpenNebula, and Ubuntu Enterprise Cloud are rated the most mature cloud infrastructures. Considering the fact that Red Hat and Ubuntu are among the main development partners in the OpenStack project, OpenStack is also a good candidate for further investigations. Like the surveys including commercial cloud infrastructures, these three can be considered outdated with respect to provided information on performance and technology, since all investigated open cloud infrastructures provided multiple updated releases per year introducing new features and technologies. For example, OpenStack introduced the concept of compute cells (i.e., multi-instantiated resource pooling) in the year 2013.[2] However, they provide a good overview of architectural concepts and features of open cloud infrastructures.

For further investigation, three commercial and two open cloud infrastructures are selected as reference for existing cloud infrastructures and are analysed on the basis of current documentation and literature. For the commercial cloud infrastructures, the following candidates are selected:

- Amazon Web Service (AWS) [5] is selected for its mature concept of distributed resource provision. Additionally, it is well documented in literature.

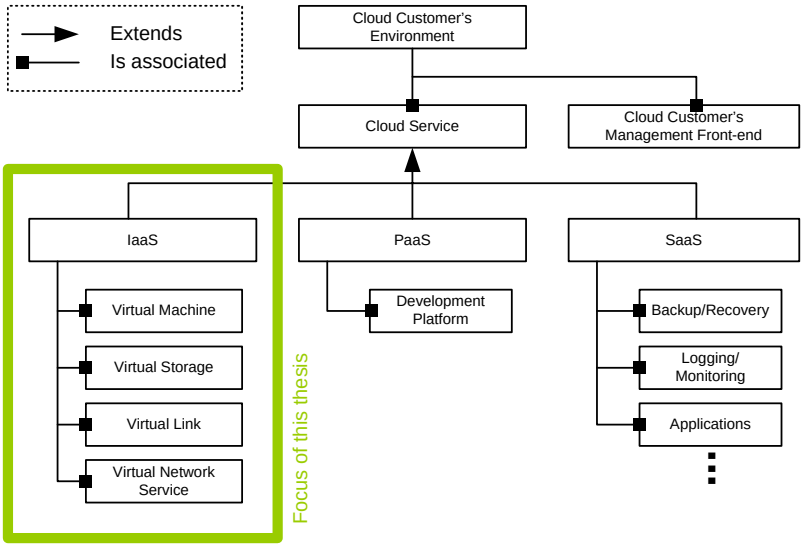
- Windows Azure [143] is selected as one of the major competitors of AWS, which unlike AWS provides both PaaS and IaaS. Windows Azure is well documented as well.

- Fujitsu Cloud IaaS Trusted Public S5 is selected because of its mature concepts for high availability, which are beyond current standards in practice.

Following the recommendation of Voras et al., OpenStack [157] and OpenNebula [156]) are selected as candidates for open cloud infrastructures.

Based on the investigation of these cloud infrastructures, a classification of IaaS computing resources is provided in the following. The results of the comparison of the five cloud infrastructures are summarised in Appendix A.

Figure 4.2: Classification of the cloud customer's environment with focus on IaaS.



4.1.2.2 Cloud customer's environment

Summarising the previous observations on cloud infrastructures and cloud services, Figure 4.2 depicts the classification of a cloud customer's environment. The environment consists of the cloud customer's management front-end and the provisioned cloud services. The cloud services are classified by the three service models, each of which contains certain types of service. For IaaS (highlighted in Figure 4.2), the service types are virtual machines ($\mathbb{VM}$), virtual storage ($\mathbb{VS}$), virtual links ($\mathbb{VL}$), and virtual network services ($\mathbb{VNS}$), which are specified in the following.

4.1.2.3 Virtual machine $\mathbb{VM}$

The class virtual machine $\mathbb{VM} := \{\overline{vm}_1, ..., \overline{vm}_n\}$ is a set of virtual machine instances $\overline{vm}_i$ with $n \in \mathbb{N}$ and $i \in \{1, ..., n\}$. The properties and components of $\mathbb{VM}$ are depicted in Figure 4.3 and explained in the following. Table A.1 of Appendix A provides an overview of the observed properties that a virtual machine instance can have in current cloud infrastructures.

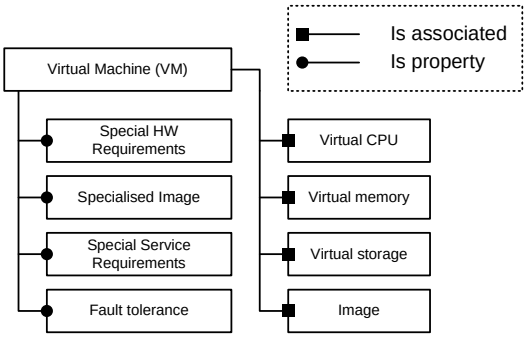
Each virtual machine instance $\overline{vm}_i \in \mathbb{VM}$ consists of computing resources classified by virtual CPU, virtual memory, virtual storage, and image (i.e., hosted operating system and installed applications).

Special hardware requirements $\mathbb{VM}|\mathbb{P}_{HW}$ The class $\mathbb{VM}|\mathbb{P}_{HW}$ contains all properties addressing direct hardware support (e.g., hyper-threading for high performance data processing,

[1] As of June 2015

[2] Compute cells were introduced with the release of OpenStack 2013.1 (Grizzly), on the Internet: `https://wiki.openstack.org/wiki/ReleaseNotes/Grizzly` (last visited: 30.06.2015)

Figure 4.3: Virtual machine classification.



hardware-based en-/decoding using GPU, and high I/O memory access) and private cloud hosting.

Table A.1 shows that all investigated cloud infrastructures provide support for special hardware requirements. However, the supported requirements can vary from private cloud hosting only to a variety of hardware-supported computing.

Specialised image $\mathbb{VM}|\mathbb{P}_{Img}$ The class $\mathbb{VM}|\mathbb{P}_{Img}$ contains all properties characterising the image (i.e., installed operating system and applications) that is running on a virtual machine instance. In general, images can be pre-defined by the cloud provider or customised by the cloud customer. Both types of image can be configured to support specific workspace environments (e.g., for office environments or development platforms) or applications (e.g., web server or database server).

All investigated infrastructures support the provisioning of pre-defined images, and three of them support customised images (cf. Table A.1). The pre-defined images vary from plain installations of operating systems to complex installations of office or development environments. Particularly, Windows Azure supports images that are used in their PaaS instances.

Special service requirements $\mathbb{VM}|\mathbb{P}_{Serv}$ The class $\mathbb{VM}|\mathbb{P}_{Serv}$ contains all properties addressing specific requirements on the operation and provisioning of the virtual machine instances (e.g., resource pooling and availability).

Four of the investigated infrastructures support availability options (i.e., high availability, scaling, provisioning, and utilisation) and two of them support resource pooling (cf. Table A.1). Windows Azure does not support special service requirements.

Fault tolerance $\mathbb{VM}|\mathbb{P}_{FT}$ The class $\mathbb{VM}|\mathbb{P}_{FT}$ contains all properties characterising fault tolerance mechanisms for virtual machine instances (i.e., backup, replication, and recovery).

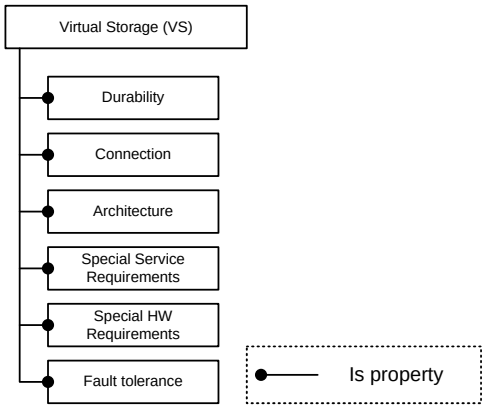
All investigated infrastructures support fault tolerance. However, the supported mechanisms are focused on backup, replication, and recovery of virtual storage instances associated with the virtual machine instance (cf. Table A.1). Four of the infrastructures explicitly support recovery mechanisms for virtual machine instances. Two infrastructures provide support services for automated recovery of virtual machine instances. In all cases of recovery, the virtual

machine instance is restarted. Only OpenNebula supports full recovery of virtual machines including virtual machine memory and non-persistent storage.

4.1.2.4 Virtual storage $\mathbb{VS}$

The class $\mathbb{VS} := \{\overline{vs}_1, ..., \overline{vs}_n\}$ is a set of virtual storage instances $\overline{vs}_i$ with $n \in \mathbb{N}$ and $i \in \{1, ..., n\}$. $\mathbb{VS}$ is atomic (i.e., has no components). A virtual storage instance $\overline{vs}_i$ describes a fixed amount of storage capacity. Table A.2 of Appendix A shows the observed properties that a virtual storage instance can have in current cloud infrastructures. The properties of $\mathbb{VS}$ are depicted in Figure 4.4 and explained in the following.

Figure 4.4: Virtual storage classification.



Durability $\mathbb{VS}|\mathbb{P}_{Dura}$ The class $\mathbb{VS}|\mathbb{P}_{Dura}$ contains the durability properties of virtual storage instances. The durability of virtual storage instances can be either non-persistent or persistent.

All investigated infrastructures provide both non-persistent and persistent virtual storage instances (cf. Table A.2). In OpenNebula, storage is freely configurable to be either persistent or non-persistent. For all other, non-persistent virtual storage instances are used to store the operating system and temporal data, and persistent virtual storage instances are used for application data and backup.

Connection $\mathbb{VS}|\mathbb{P}_{Con}$ The class $\mathbb{VS}|\mathbb{P}_{Con}$ contains the connection properties of virtual storage instances with respect to the virtual machine instance. The connection of virtual storage instances can be either local (i.e., co-located with the virtual machine instance) or remote.

All investigated infrastructures support remotely located virtual storage instances (cf. Table A.2). Three of them also support locally connected virtual storage instances.

Architecture $\mathbb{VS}|\mathbb{P}_{Arch}$ The class $\mathbb{VS}|\mathbb{P}_{Arch}$ contains the architecture properties of virtual storage instance. The architecture of virtual storage instances can be either block storage (i.e., organised by blocks of binary data) or object storage (i.e., organised by using references on data objects).

All investigated infrastructures support block storage for virtual storage instances (cf. Table A.2). Four of them also support object storage.

Special service requirements $\mathbb{VS}|\mathbb{P}_{Serv}$ The class $\mathbb{VS}|\mathbb{P}_{Serv}$ contains all properties addressing specific requirements on the operation and provisioning of virtual storage instances (e.g., availability, access, and distribution).

All investigated infrastructures support different types of service requirement for virtual storage instances (cf. Table A.2). Content distribution and access-related properties are supported by four infrastructures. The Fujitsu Cloud infrastructure focuses on the support of availability- and security-related properties.

Special hardware requirements $\mathbb{VS}|\mathbb{P}_{HW}$ The class $\mathbb{VS}|\mathbb{P}_{HW}$ contains all properties addressing direct hardware support (e.g., SSD-support) and private cloud hosting.

Two of the investigated infrastructures do not support special hardware requirements for virtual storage instances (cf. Table A.2). AWS provides SSD-support and for OpenStack, resource pooling is a planned property. The Fujitsu cloud infrastructure provides private cloud computing by hosting on dedicated servers. OpenNebula supports customisable hardware support by introducing three types of hardware resource pooling concept.

Fault tolerance $\mathbb{VS}|\mathbb{P}_{FT}$ The class $\mathbb{VS}|\mathbb{P}_{FT}$ contains all properties characterising fault tolerance mechanisms for virtual storage instances (i.e., backup, replication, and recovery).

All of the investigated infrastructures support fault tolerance mechanisms for virtual storage instances (cf. Table A.2). There are mechanisms supporting both single-site and distributed backup and recovery. In all cases of recovery, the virtual machine instance has to be stopped to recover associated virtual storage instances and both the virtual machine's memory and any non-persistent storage is lost. Windows Azure provides a support service for the automated recovery of virtual storage instances.

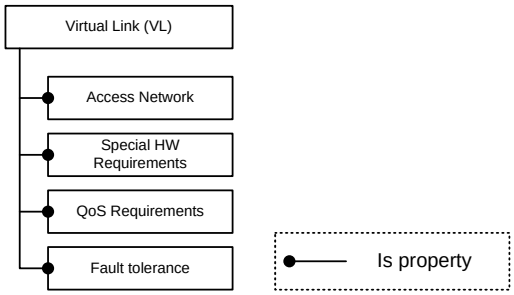
4.1.2.5 Virtual link $\mathbb{VL}$

The class virtual link $\mathbb{VL}:=\{\overline{vl}_1,...,\overline{vl}_n\}$ is a set of virtual link instances $\overline{vl}_i$ with $n \in \mathbb{N}$ and $i \in \{1,...,n\}$. A virtual link instance $\overline{vl}_i$ describes a connection between two end-systems (e.g., a virtual machine instance and a cloud customer's device). Table A.3 of Appendix A shows the observed properties that a virtual link instance can have in current cloud infrastructures. The properties of $\mathbb{VL}$ are depicted in Figure 4.5 and explained in the following.

Access network $\mathbb{VL}|\mathbb{P}_{Access}$ The class $\mathbb{VL}|\mathbb{P}_{Access}$ contains the access network properties of a virtual link instance. These properties are given if a virtual link instance is used to provide access to the virtual resource instances. The access network can be either a publicly available or privately used connection of the cloud customer.

All investigated infrastructures provide both public and private access networks (cf. Table A.3). Two infrastructures provide a private access network by default. All other infrastructures provide a public access network by default.

Figure 4.5: Virtual link classification.



Special hardware requirements $\mathbb{VL}|\mathbb{P}_{HW}$ The class $\mathbb{VL}|\mathbb{P}_{HW}$ contains all properties addressing direct hardware support (e.g., specialised access router).

Three of the investigated infrastructures do not support special hardware requirements for virtual link instances (cf. Table A.3). The two other infrastructures support the utilisation of specialised access routers.

QoS requirements $\mathbb{VL}|\mathbb{P}_{QoS}$ The class $\mathbb{VL}|\mathbb{P}_{QoS}$ contains all properties addressing QoS requirements of the virtual link instances (e.g., bandwidth, delay, and packet loss rate).

All investigated infrastructures support QoS requirements for virtual link instances (cf. Table A.3). Three infrastructures support high availability. AWS supports low latency and high performance networking.

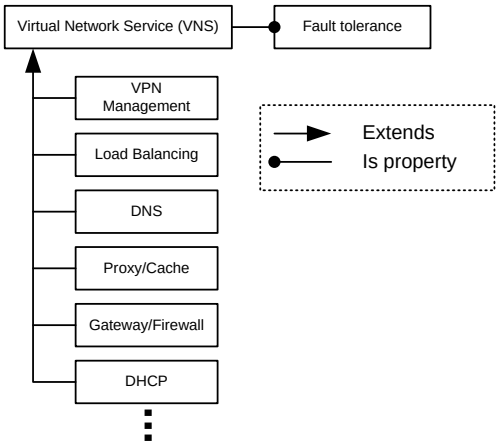
Fault tolerance $\mathbb{VL}|\mathbb{P}_{FT}$ The class $\mathbb{VL}|\mathbb{P}_{FT}$ contains all properties characterising fault tolerance mechanisms for virtual link instances (e.g., redundancy and error correction).

Two of the investigated infrastructures provide support for redundancy (cf. Table A.3). Windows Azure provides 2-out-of-2 redundancy (2oo2) for private access networks (i.e., network connect only fails if both links fail at the same time). The Fujitsu Cloud infrastructure provides by default redundancy on all network devices a Local Area Network (LAN) cabling. For AWS, there is – to the best of the author's knowledge – no publicly available information on fault tolerance available. It is reasonable to assume that all infrastructure uses basic error detection and correction mechanisms provided by standard communication protocols (e.g., checksums and acknowledgements).

4.1.2.6 Virtual network service $\mathbb{VNS}$

The class virtual network service $\mathbb{VNS} := \{\overline{vns}_1, ..., \overline{vns}_n\}$ is a set of virtual network service instances $\overline{vns}_i$ with $n \in \mathbb{N}$ and $i \in \{1, ..., n\}$. A virtual network service instance $\overline{vns}_i$ is a virtualised application in communication networks (e.g., Virtual Private Network (VPN) management, Dynamic Host Configuration Protocol (DHCP), and load balancing). Table A.4 of Appendix A shows examples of observed types of virtual network service instance in current cloud infrastructures and their properties. The types and properties of $\mathbb{VNS}$ are depicted in Figure 4.6.

Figure 4.6: Virtual network service classification.



All investigated cloud infrastructures provide a variety of virtual network services (cf. Table A.4). While most services are optional, DHCP is provided by default in all infrastructures, since automatic network configuration is important for scalability and self-servicing.

Fault tolerance $\mathbb{VNS}|\mathbb{P}_{FT}$ The class $\mathbb{VNS}|\mathbb{P}_{FT}$ contains all properties characterising fault tolerance mechanisms for virtual network service instances (e.g., redundancy and recovery).

For any infrastructure, there is no publicly available information on the fault tolerance of the virtual network services (cf. Table A.4). However, it is reasonable to assume that fault tolerance mechanisms are implemented for the fail-safe operation of automatically provided network configurations.

4.1.2.7 Virtual resource $\mathbb{VR}$

For completeness, the class virtual resource $\mathbb{VR}$ is defined as the union of all virtual resource classes that were defined previously: $\mathbb{VR} := \mathbb{VM} \cup \mathbb{VS} \cup \mathbb{VL} \cup \mathbb{VNS}$.

4.1.3 Classification of hardware resources

The backbone of cloud infrastructures is the hosting sites (i.e., data centres) providing the required hardware resources to operate and provide cloud services.

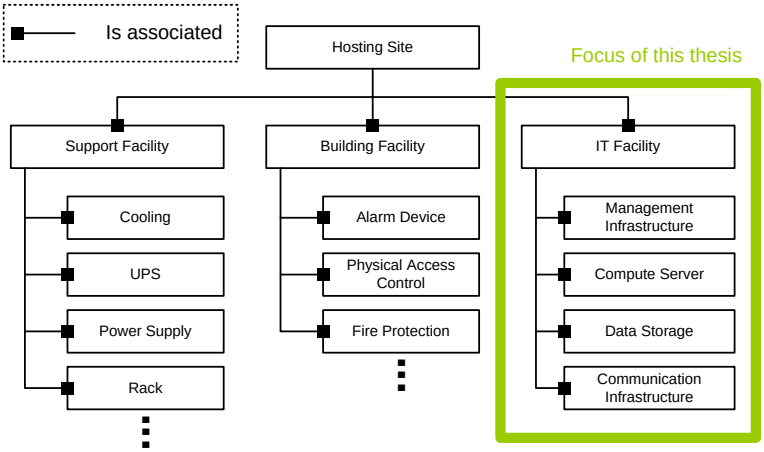
A hosting site is usually located in a building specifically designed for operating hardware resources. It contains IT facilities representing the hardware resources that can be classified by network (i.e., communication infrastructure), services (of the communication infrastructure), computation (i.e., server), storage, and management (cf. Cisco VMDC Layers [44]). Beside the hardware resources, there are support facilities (e.g., cooling, UPS, power supply, and racks) that are necessary for operating the hardware resources. Additionally, there are building facilities (e.g., alarm devices, physical access control, and fire protection) that are necessary for the operation, maintenance, and protection of the building itself.

To gain a better view of hardware resources, it is necessary to understand how they are structured and operated in practice. Here, the design guide for Virtualized Multiservice Data Center (VMDC) [44] and the Common Information Model (CIM) of the Distributed Management Task Force (DMTF) [57] provide representative and comprehensive information. The design guide for VMDC provides detailed information on state of the art data centre infrastructure set-up and operation as recommended by CISCO,[1] which are one of the leading companies for the wired and wireless LAN access infrastructures [200]. The CIM is a standard for management information for IT systems and covers a systematic description of IT systems and their interaction. Both references are used in the following to derive a classification of hardware resources provided by hosting sites in the context of cloud computing.

4.1.3.1 Hosting site $\mathbb{HS}$

Summarising the previous observations, the class hosting site $\mathbb{HS}$ is defined as a set of support facilities, building facilities, and IT facilities. Figure 4.7 depicts the components of the class hosting site. The hardware resources are represented by the IT facilities (highlighted in Figure 4.7) and are classified by management infrastructure, compute server, data storage, and communication infrastructure. Each of these classes is specified in the following.

Figure 4.7: Classification of hosting sites with a focus on the IT facility.



4.1.3.2 Management infrastructure $\mathbb{MI}$

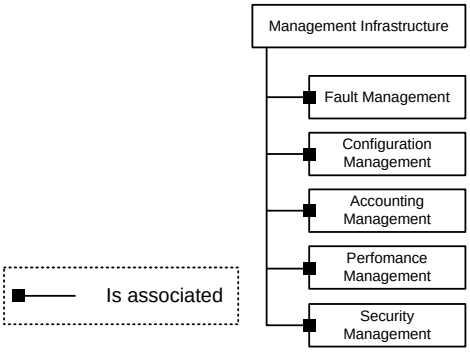
The class management infrastructure $\mathbb{MI} := \{\overline{mi}_1, \ldots, \overline{mi}_n\}$ is a set of management infrastructure instances $\overline{mi}_i$ with $n \in \mathbb{N}$ and $i \in \{1, \ldots, n\}$. The components of $\mathbb{MI}$ are depicted in Figure 4.8.

A management infrastructure instance $\overline{mi}_i$ is an automated management function classified by fault management, configuration management, accounting management, performance man-

[1] Cisco Systems, Inc., in the Internet: http://www.cisco.com (last visited: 30.06.2015).

agement, and security management (cf. CCITT recommendation on management functions [197]). The management function classes are specified as follows.

Figure 4.8: Classification of the management infrastructure.



- **Fault management** $\mathbb{FMF}$: The class $\mathbb{FMF}$ is a set of functions for detecting, isolating and correcting abnormal operation of the IT facilities (cf. [197], pp. 15 et. seqq.). In particular, this includes the functions for backup $\overline{fmf}_{Bak}$, replication $\overline{fmf}_{Repl}$, and recovery $\overline{fmf}_{Recover}$ of IT facilities.
- **Configuration management** $\mathbb{CM}$: The class $\mathbb{CM}$ is a set of functions for controlling, identifying, collecting data from, and providing data to IT facilities (cf. [197], pp. 37 et. seqq.).
- **Accounting management** $\mathbb{AMF}$: The class $\mathbb{AMF}$ is a set of functions for measuring usage and determining costs of IT facilities, and charging customers, i.e., cloud providers (cf. [197], pp. 53 et. seqq.).[1]
- **Performance management** $\mathbb{PMF}$: The class $\mathbb{PMF}$ is a set of functions for evaluating and reporting on the behaviour and effectiveness of IT facilities (cf. [197], pp. 5 et. seqq.).
- **Security management** $\mathbb{SMF}$: The class $\mathbb{SMF}$ is a set of functions for preventing and detecting security incidents, containment and recovery of IT facilities after the occurrence of security incidents, and security administration (cf. [197], pp. 62 et. seqq.).

4.1.3.3 Compute server $\mathbb{CS}$

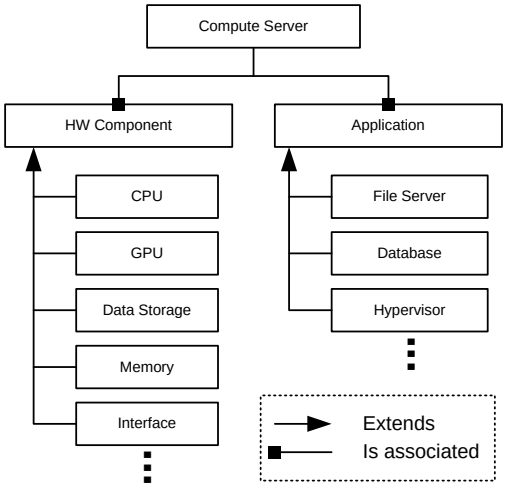
The class compute server $\mathbb{CS} := \{\overline{cs}_1, ..., \overline{cs}_n\}$ is a set of compute server instances $\overline{cs}_i$ with $n \in \mathbb{N}$ and $i \in \{1, ..., n\}$. The components of $\mathbb{CS}$ are depicted in Figure 4.9.

A compute server instance $\overline{cs}_i$ is a physical server (organised in racks or as stand-alone) consisting of hardware components and of hosted applications, which are specified as follows.

[1] From the point of view of a hosting site, the cloud provider is a customer who requests hardware resources for operating a cloud infrastructure. Usually, there is no direct contact between a hosting site and a cloud customer.

- **Hardware component** $\mathbb{HW}$: The class $\mathbb{HW}$ is a set of different types of hardware component (e.g., CPU, GPU, data storage, memory, and interfaces) a physical server is built of. Each hardware component can be used to provide a specific amount of computation resource (e.g., CPU cycles and memory size).
- **Application** $\mathbb{APP}$: The class $\mathbb{APP}$ is a set of application types (e.g., file server, database, and hypervisor) that are hosted on the compute server. Each application type can have multiple, independent instances that can be utilised for providing cloud services (e.g., virtual network services and virtual machines).

Figure 4.9: Compute server classification.



4.1.3.4 Data storage $\mathbb{DS}$

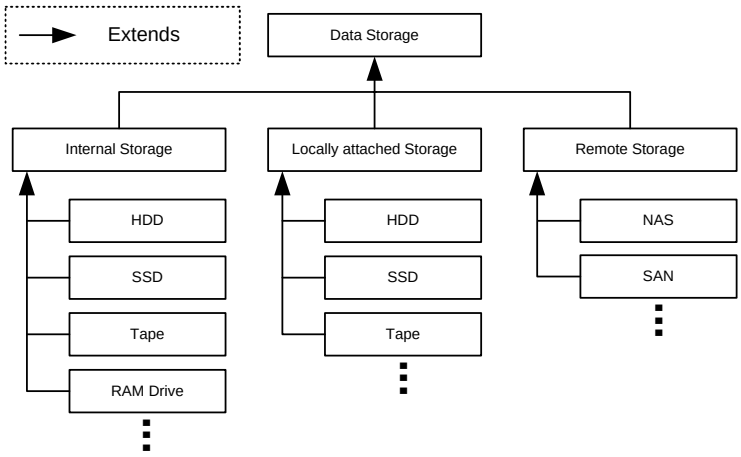
The class data storage $\mathbb{DS} := \{\overline{ds}_1, ..., \overline{ds}_n\}$ is a set of data storage instances $\overline{ds}_i$ with $n \in \mathbb{N}$ and $i \in \{1, ..., n\}$. As depicted in Figure 4.10, a data storage instance $\overline{ds}_i$ (i.e., physical data storage) can be classified by type into internal storage (of the compute server), locally attached storage (of the compute server), and remote storage (provided via communication infrastructure), which are specified in the following.

Internal data storage $\mathbb{IDS}$ The class $\mathbb{IDS}$ is a set of data storage instances $\overline{ds} \in \mathbb{DS}$ that are located inside a physical server chassis. $\mathbb{IDS}$ can be classified into different types of internal data storage device (e.g., hard disk drive (HDD), solid-state drive (SSD), and tape).

Locally attached data storage $\mathbb{ADS}$ The class $\mathbb{ADS}$ is a set of data storage instances $\overline{ds} \in \mathbb{DS}$ that are locally attached to a physical server (e.g., via Universal Serial Bus (USB) or via Serial Advanced Technology Attachment (SATA) for Direct Attached Storage (DAS)). $\mathbb{ADS}$ can be classified into different types of attached data storage device (e.g., HDD, SSD, and tape).

Remote data storage $\mathbb{RDS}$ The class $\mathbb{RDS}$ is a set of data storage instances $\overline{ds} \in \mathbb{DS}$ that are available via a network connection (e.g., via LAN or Storage Area Network (SAN)). $\mathbb{RDS}$ can be classified into different types of remote data storage type (e.g., network-attached storage (NAS) and SAN).

Figure 4.10: Data storage classification.



Remark 4.1 (Fault tolerance of data storage) *For reasons of fault tolerance, data storage is usually managed in multiple layers of synchronised replications. An example of a two-layer replication would be a first-level replication for storage protection, and a second-level replication for disaster recovery (cf. VMDC service tiers [44]). For a data storage instance* $\overline{ds} \in \mathbb{DS}$*, this replication is modelled as follows.*

$$(\overline{ds}, \overline{ds}|\overline{fmf}_{Repl}, \overline{ds}|\overline{fmf}_{Repl}|\overline{fmf}_{Repl}) \text{ with } \overline{fmf}_{Repl} \in \mathbb{FMF} .$$

For the same instance, a 1-layer replication with 2oo2 redundancy is modelled as follows.

$$(\overline{ds}, \overline{ds}|\overline{fmf}_{Repl}, \overline{ds}|\overline{fmf}_{Repl}') \text{ with } \overline{fmf}_{Repl} \in \mathbb{FMF} .$$

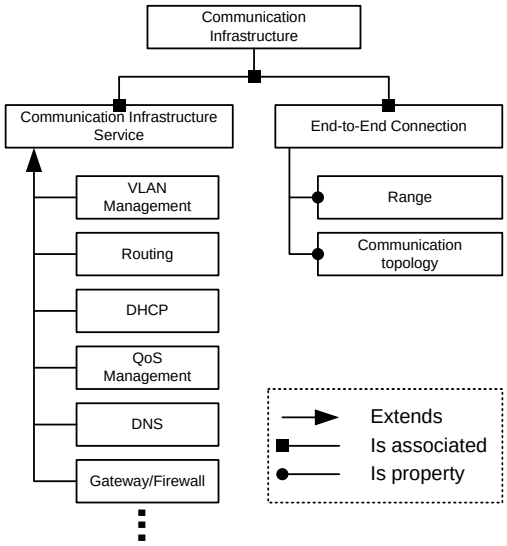
4.1.3.5 Communication infrastructure $\mathbb{CI}$

The class communication infrastructure $\mathbb{CI} := \{\overline{ds}_1, ..., \overline{ds}_n\}$ is a set of communication infrastructure instances $\overline{ds}_i$ with $n \in \mathbb{N}$ and $i \in \{1, ..., n\}$.

A communication infrastructure instance $\overline{ds}_i$ consists of all elements of a network infrastructure that are necessary for the communication between compute system instances and data storage instances. These are communication services (e.g., Virtual Local Area Network (VLAN) management, QoS management, and routing) and service endpoints required for creating end-to-end connections between compute system instances and data storage instances (cf. CIM network schema [57]).

The components of $\mathbb{CI}$ are depicted in Figure 4.10 and specified in the following:

Figure 4.11: Communication infrastructure classification.



Communication infrastructure service $\mathbb{CIS}$ The class $\mathbb{CIS}$ is a set of service instances and can be classified by different types of service (e.g., VLAN management, routing, and DHCP). Each service instance is an application instance (e.g., DHCP server) hosted on a compute server connected to the communication infrastructure.

End-to-end connection $\mathbb{CON}$ The class $\mathbb{CON}$ is a set of end-to-end connection instances established between two or more compute server instances and/or data storage instances. For example, an end-to-end connection instance $\overline{con}$ between the compute server instances $\overline{cs_1}$ and $\overline{cs_2}$ is described as follows.

$$\overline{con}(\overline{cs_1}, \overline{cs_2}) := \overline{con} \sim (\overline{cs_1}, \overline{cs_2}) \ .$$

An end-to-end connection instance $\overline{con}$ has the following properties.

- **Range** $\mathbb{CON}|\mathbb{P}_{Range}$: The property $\mathbb{CON}|\mathbb{P}_{Range} := \{\overline{con}|\mathbb{P}_{Range,In}, \overline{con}|\mathbb{P}_{Range,Ex}\}$ describes whether the connection endpoints of $\overline{con}$ are located within the hosting site ($\overline{con}|\mathbb{P}_{Range,In}$), or one or more connection endpoints of $\overline{con}$ are located outside of the hosting site ($\overline{con}|\mathbb{P}_{Range,Ex}$).
- **Communication topology** $\mathbb{CON}|\mathbb{P}_{Top}$: The property $\mathbb{CON}|\mathbb{P}_{Top}$ describes the topology of the communication, for example, one-to-one communication ($\overline{con}|\mathbb{P}_{Top,1-1}$) or one-to-many ($\overline{con}|\mathbb{P}_{Top,1-*}$). For example, an 1-to-n communication between the compute server instance $\overline{cs_1}$ and the group of compute server instances $\overline{cs_2}, ..., \overline{cs_n}$ (with $n \in \mathbb{N}$) is described as follows.

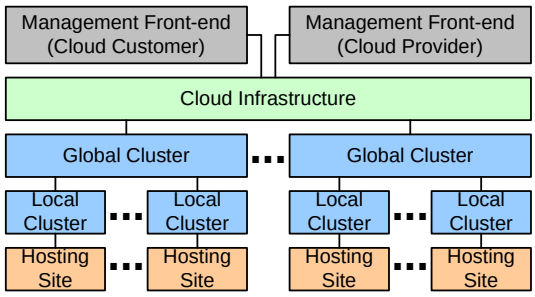
$$\overline{con}(\overline{cs}_1, (\overline{cs}_2, ..., \overline{cs}_n)) := \overline{con} \sim (\overline{cs}_1, (\overline{cs}_2, ..., \overline{cs}_n))$$
$$\text{with } \overline{con} \in \mathbb{CON} \text{ and } \overline{con}|\mathbb{P}_{Top,1-n} \in \mathbb{CON}|\mathbb{P}_{Top} \ .$$

4.1.4 Cloud infrastructure

For service orchestration at the cloud provider, computing resources are composed to cloud services that are provided to the cloud customers. The cloud management process describes the management and composition of virtual resource instances at the cloud provider. It covers the creation, migration, and destruction of virtual resource instances, the pooling of compute resources provided by the hosting sites, and the composition of virtual resource instances to cloud services usable by the cloud customer. It also provides interfaces to the management front-ends of cloud customers and cloud providers, for example the Open Cloud Computing Interface (OCCI) [142] and the Cloud Infrastructure Management Interface (CIMI) [54].

Cloud infrastructures are usually organised in local resource clusters, which are coordinated by superordinate global clusters. Figure 4.12 depicts the cloud infrastructure with the management front-ends and the underlying clusters and hosting sites. The global clusters pool the local clusters, (e.g., 'Regions' in OpenStack and AWS or 'oZones' in OpenNebula). The local clusters typically cover the resources of a single hosting site (e.g., 'availability zones' of AWS or 'Islands' of the Fujitsu cloud infrastructure) and can be hierarchically structured into smaller subclusters, for example into 'cells' in OpenStack or 'cluster' and 'virtual data centres' in OpenNebula. The subclusters help to organise hardware resources and are usually located at the same hosting site as is the superordinate local cluster. Without loss of generality, subclusters are considered in the following as a part of the local cluster, and therefore as having the same location and same entity responsible for its operation.

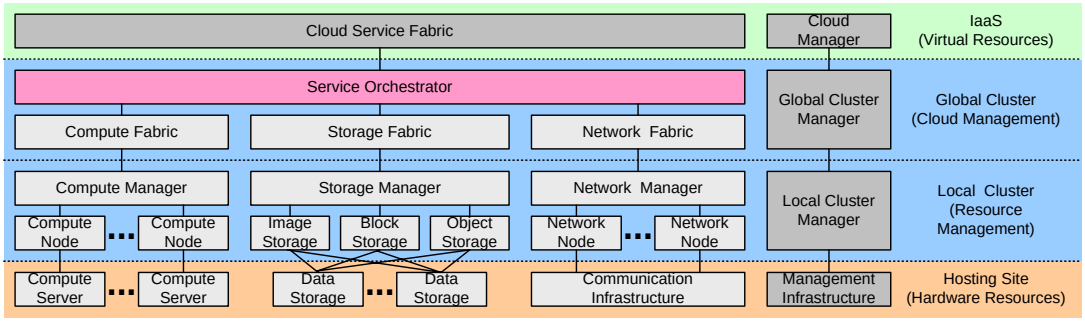
Figure 4.12: Components of the cloud management structure.



The cloud management process operates on the cloud management infrastructure. Figure 4.13 depicts the components of the cloud management process, their location within the cloud management infrastructure, and their control hierarchy as they can be observed in existing cloud infrastructures [204]. At the top there is the cloud service fabric providing the cloud services to the customer and the cloud manager controlled by the cloud provider. Each global cluster consists of a service orchestrator which creates cloud services by composing virtual resources, the fabrics of the virtual resources, and the global cluster manager. Each local cluster consists of the resource manager controlling the virtual resources provided by the underlying hosting site.

The components of both the cloud management infrastructure and the cloud management process are specified in the following.

Figure 4.13: Components of the cloud management process.



4.1.4.1 Cloud infrastructure $\mathbb{CL}$

The class $\mathbb{CL}$ defines a set of cloud infrastructure instances. Each cloud infrastructure instance consists of a cloud manager instance, a cloud service fabric instance, and a set of global cluster instances, which all are specified in the following.

Cloud manager $\mathbb{CM}$ The class $\mathbb{CM}$ defines a set of cloud manager instances that are responsible for coordinating the operation and interaction of all components of the cloud infrastructure instance (i.e., cloud service fabric instance and global cluster instances). A cloud manager instance is controlled by the cloud management front-end of the cloud provider.

Cloud service fabric $\mathbb{CSF}$ The class $\mathbb{CSF}$ defines a set of cloud service fabric instances that are responsible for coordinating the creation and provisioning of cloud services (i.e., virtual resources for IaaS). Each cloud service fabric instance is controlled by the cloud management front-end of the cloud customer and provides cloud service instances operated by the global clusters. A cloud service fabric instance $\overline{csf} \in \mathbb{CSF}$ is defined a function $\overline{csf} : \mathcal{P}(\mathbb{VR}) \rightarrow \mathcal{P}(\mathbb{VR})^{|\mathbb{GC}|}$ that partitions the input set $\overline{VR} \in \mathcal{P}(\mathbb{VR})$ on the global clusters, i.e.,

$$\overline{csf}(\overline{VR}) := (\overline{VR}_{\overline{gc}_1}, ..., \overline{VR}_{\overline{gc}_n}) \text{ with } n = |\mathbb{GC}| \text{ and } \overline{gc}_i \in \mathbb{GC} \text{ for } i \in \{1, ..., n\} \ .$$

4.1.4.2 Global cluster $\mathbb{GC}$

The class $\mathbb{GC}$ defines a set of global cluster instances representing regions or zones grouping local infrastructure elements within the cloud infrastructure. A global cluster instance consists of global cluster manager instance, a service orchestrator instance, fabric instances for compute, storage and network resources, and a set of local clusters, which are all specified in the following.

Global cluster manager $\mathbb{GCM}$ The class $\mathbb{GCM}$ defines a set of global cluster manager instances that are responsible for coordinating the operation and interaction of all components of the global cluster infrastructure instance (i.e., service orchestrator instance, fabric instances,

and local cluster instances). A global cluster manager instance is controlled by the cloud manager instance of the superordinate cloud infrastructure instance.

Service orchestrator $\mathbb{SO}$ The class $\mathbb{SO}$ defines a set of service orchestrator instances that are responsible for composing the virtual resources to cloud service instances. Each service orchestrator instance requests virtual resources from the fabric instances and provides cloud service instances to the service fabric instance of the superordinate cloud infrastructure instance. A service orchestrator instance $\overline{so} \in \mathbb{SO}$ is defined as a function $\overline{so} : \mathcal{P}(\mathbb{VR}) \rightarrow \mathcal{P}(\mathbb{VM}) \times \mathcal{P}(\mathbb{VS}) \times \mathcal{P}(\mathbb{VL} \cup \mathbb{VNS})$ that partitions the input set $\overline{VR} \in \mathcal{P}(\mathbb{VR})$ on the global cluster's fabrics, i.e.,

$$\overline{so}(\overline{VR}) := (\overline{VM}, \overline{VS}, \overline{VN}) \text{ with } \overline{VM} \in \mathbb{VM}, \overline{VS} \in \mathbb{VS} \text{ and } \overline{VN} \in \mathbb{VL} \cup \mathbb{VNS} \ .$$

Fabrics $\mathbb{CF}$, $\mathbb{SF}$, $\mathbb{NF}$ The class $\mathbb{CF}$ defines a set of compute fabric instances that are responsible for coordinating the creation and provisioning of virtual machine instances. Each compute fabric instance is controlled by the global cluster manager instance and provides virtual machine instances to the service orchestrator. A compute fabric instance $\overline{cf} \in \mathbb{CF}$ is defined as a function $\overline{cf} : \mathcal{P}(\mathbb{VM}) \rightarrow \mathcal{P}(\mathbb{VM})^{|\overline{LC}_{\overline{gc}}|}$ that partitions the input set $\overline{VM} \in \mathcal{P}(\mathbb{VM})$ on the set of local clusters $\overline{LC}_{\overline{gc}} \subseteq \mathbb{LC}$ of the global cluster $\overline{gc} \in \mathbb{GC}$, i.e.,

$$\overline{cf}(\overline{VM}) := (\overline{VM}_{\overline{lc}_1}, ..., \overline{VM}_{\overline{lc}_n}) \text{ with } n = \left|\overline{LC}_{\overline{gc}}\right| \text{ and } \overline{lc}_i \in \overline{LC}_{\overline{gc}} \text{ for } i \in \{1, ..., n\} \ .$$

The class $\mathbb{SF}$ is defined analogously for the creation and provision of virtual storage instances, and $\mathbb{NF}$ is defined analogously for virtual link instances and virtual network service instances, respectively.

4.1.4.3 Local cluster $\mathbb{LC}$

The class $\mathbb{LC}$ defines a set of local cluster instances representing local infrastructure elements located at a single hosting site. A local cluster instance consists of a local cluster manager instance, manger instances for compute, storage and network resources, and compute, storage and network resource instances, which all are specified in the following.

Local cluster manager $\mathbb{GCM}$ The class $\mathbb{GCM}$ defines a set of local cluster manager instances that are responsible for coordinating the operation and interaction of all components of the local cluster infrastructure instance (i.e., resource manager instances and resources) and interacts with the management infrastructure instances of the underlying hosting site. A local cluster manager instance is controlled by the global cluster instance of the superordinate global cluster instance.

Resource manager $\mathbb{CM}$, $\mathbb{SM}$, $\mathbb{NM}$ The class $\mathbb{CM}$ defines a set of compute manager instances that control the creation, operation, and destruction of virtual machine instances, which is performed by the compute node instances. A compute manager instance $\overline{cm} \in \mathbb{CM}$ is defined

as a function $\overline{cm} : \mathcal{P}(\mathbb{VM}) \to \mathcal{P}(\mathbb{VM})^{\left|\overline{CN}_{\overline{lc}}\right|}$ that partitions the input set $\overline{VM} \in \mathcal{P}(\mathbb{VM})$ on the set of compute nodes instances $\overline{CN}_{\overline{lc}} \subseteq \mathbb{CN}$ of the local cluster $\overline{lc} \in \mathbb{LC}$, i.e.,

$$\overline{cm}(\overline{VM}) := (\overline{VM}_{\overline{cn}_1}, ..., \overline{VM}_{\overline{cn}_n}) \text{ with } n = \left|\overline{CN}_{\overline{lc}}\right| \text{ and } \overline{cn}_i \in \overline{CN}_{\overline{lc}} \text{ for } i \in \{1, ..., n\} \ .$$

The class $\mathbb{SM}$ is defined analogously for virtual storage instances in conjunction with storage pool instances, and the class $\mathbb{NM}$ analogously for virtual link and virtual network service instances in conjunction with network node instances.

Compute node $\mathbb{CN}$ The class $\mathbb{CM}$ defines a set of compute node instances which are providing the virtual machine instances. Each compute node instance represents a compute server of the underlying hosting site instance and operates a hypervisor instance that can create, operate, and destroy virtual machine instances. Each compute node instance is controlled by the compute manager instance of the same local cluster instance. A compute node instance $\overline{cn} \in \mathbb{CN}$ is defined as a function $\overline{cn} : \mathcal{P}(\mathbb{VM}) \to \mathbb{CS}$ that assigns the input set $\overline{VM} \in \mathcal{P}(\mathbb{VM})$ to the set of compute server instances $\overline{CS}_{\overline{lc}} \subseteq \mathbb{CS}$ at the hosting site associated with the local cluster $\overline{lc} \in \mathbb{LC}$, i.e.,

$$\overline{cn}(\overline{VM}) := \overline{cs} \text{ with } \overline{cs} \in \overline{CS}_{\overline{lc}} \ .$$

Storage pools $\mathbb{OS}$, $\mathbb{BS}$, $\mathbb{IS}$ The class $\mathbb{OS}$ is defined as a set of object storage pool instances providing virtual storage which is structured in objects. Each object storage pool instance consists of one or more data storage instances of the underlying hosting site instance and is controlled by the storage manager instance of the same local cluster instance. An object storage pool instance $\overline{os} \in \mathbb{OS}$ is defined as a function $\overline{os} : \mathcal{P}(\mathbb{VS}) \to \mathcal{P}(\mathbb{DS})$ that assigns the input set $\overline{VS} \in \mathcal{P}(\mathbb{VS})$ to the set of data storage instances $\overline{DS}_{\overline{lc}} \subseteq \mathbb{DS}$ at the hosting site associated with the local cluster $\overline{lc} \in \mathbb{LC}$, i.e.,

$$\overline{os}(\overline{VS}) := \overline{DS} \text{ with } \overline{DS} \subseteq \overline{DS}_{\overline{lc}} \text{ and } \forall \overline{vs} \in \overline{VS} \ \exists \overline{ds} \in \overline{DS} : \overline{os}(\{\overline{vs}\}) = \left\{\overline{ds}\right\} \ .$$

The class $\mathbb{BS}$ is defined analogously for block storage. The class $\mathbb{IS}$ is defined analogously as a set of image storage pool instances (a.k.a. image repository) that are responsible for providing image instances for the creation of virtual machine instances.

Network node $\mathbb{NN}$ The class $\mathbb{NN}$ is defined as a set of network node instances that are responsible for providing and operating virtual link and virtual network service instances. Each network node instance represents an element of the communication infrastructure instance of the underlying hosting site instance and is controlled by the network manager instance of the same local cluster instance. A network node instance $\overline{nn} \in \mathbb{NN}$ is defined as a function $\overline{nn} : \mathcal{P}(\mathbb{VL} \cup \mathbb{VNS}) \to \mathcal{P}(\mathbb{CIS} \cup \mathbb{CON})$ that assigns the input set $\overline{VN}_{\overline{lc}} \in \mathcal{P}(\mathbb{VL} \cup \mathbb{VNS})$ to the output set $\overline{COM}_{\overline{hs}} \subseteq \mathcal{P}(\mathbb{CIS} \cup \mathbb{CON})$ at the hosting site $\overline{hs} \in \mathbb{HS}$ associated with the local cluster

$\overline{lc} \in \mathbb{LC}$, i.e.,

$$\overline{nn}(\overline{VN}) := \overline{COM} \quad \text{with } \overline{COM} \subseteq \overline{COM}_{\overline{lc}} ,$$
$$\forall \overline{vl} \in \overline{VN} \cap \mathbb{VL}\ \exists \overline{con} \in \overline{COM} \cap \mathbb{CON} : \overline{nn}(\{\overline{con}\}) = \{\overline{vl}\} \text{ , and}$$
$$\forall \overline{vns} \in \overline{VN} \cap \mathbb{VNS}\ \exists \overline{cis} \in \overline{COM} \cap \mathbb{CIS} : \overline{nn}(\{\overline{vns}\}) = \{\overline{cis}\} .$$

4.1.4.4 Cloud management process $\overline{cmp}$

The cloud management process is defined as the function $\overline{cmp} : \mathcal{P}(\mathbb{VR}) \times \mathcal{P}(\mathbb{HW})$ that maps requested virtual resource instances to the necessary hardware resource instances. The cloud management process can be described as the composition $\circ$ of its components' functions, i.e., for the functions f and $g_1, ...g_n$, defined as follows,

$$f : X \to Y_1 \times ... \times Y_n \text{ , } f(x) := (y_1, ..., y_n) \text{ and } g_i : Y_i \to Z_i \text{ , } g(y) := z \quad \text{with } n \in \mathbb{N} \text{ and } i \in \{1, ..., n\} ,$$

the composition $f \circ (g_1, ..., g_n)$ is defined:

$$f \circ (g_1, ..., g_n) : X \to Z_1 \times ... \times Z_n \text{ , } f \circ (g_1, ..., g_n)(x) := (f\big|^{Y_1} \circ g_1(x), ..., f\big|^{Y_n} \circ g_n(x)) \quad \text{with } f\big|^{Y_i} : X \to Y_i \text{ , } f\big|^{Y_i}(x) = y_i .$$

The composition of single-dimensional functions is represented for $n = 1$. Further, the composition of multi-dimensional functions $f : X \to Y_1 \times ... \times Y_n$ and $g : Y_1 \times ... \times Y_n \to Z$ with $n \in \mathbb{N}$ can be described as the composition $f \circ (g_1, ..., g_n) \circ h$ with $g_i : Y_i \to Z$ decomposition of g for $i \in \{1, ..., n\}$ and $h : Z^n \to Z$ recomposition of g.

An example of a management process instance $\overline{cmp}$ that is requesting virtual machine instances from a compute node instance would be (assuming that there is only one instance of each global cluster, local cluster and compute node):

$$\overline{cmp} = \overline{csf} \circ \overline{so} \circ \overline{cf} \circ \overline{cm} \circ \overline{cn}$$
$$\text{with } \overline{csf} \in \mathbb{CSF}, \overline{so} \in \mathbb{SO}, \overline{cf} \in \mathbb{CF}, \overline{cm} \in \mathbb{CM}, \text{ and } \overline{cn} \in \mathbb{CN} .$$

Table 4.1 lists examples of the resource mapping by the cloud management process. The first column describes the input (i.e., virtual resource instances) of the cloud management process. The second column shows the components of the composed cloud management process. The third column lists the result (i.e., hardware resource instances) of the cloud management process. In this manner, it is possible to describe the mapping of any virtual resource instance (and its properties) to its assigned hardware resources as a result of the cloud management process.

Consequently, the model describes all relevant entities and relations of an IaaS cloud infrastructure. Virtual resources, hardware resources and the cloud management process (linking virtual resources and hardware resources with each other) are covered. By using the formal notation of an ontology, the model forms a basis for the information flow analysis in Chapter 5. For its construction, five representative cloud infrastructures and existing standards and best

Table 4.1: Mapping virtual resources to hardware resources and the according cloud management process

Virtual resources $\overline{VR} \in \mathcal{P}(\mathbb{VR})$	**Cloud management process**[a] $\overline{cmp} : \mathcal{P}(\mathbb{VR}) \times \mathcal{P}(\mathbb{HW})$	**Hardware resources** $\overline{cmp}(\overline{VR}) = \overline{HW} \in \mathcal{P}(\mathbb{HW})$
$\{\overline{vm}\}$ with $\overline{vm} \in \mathbb{VM}$ (standard VM instance without virtual storage)	$\overline{cmp} := \overline{csf} \circ \overline{so}_{\overline{glo}} \circ \overline{cf}_{\overline{glo}} \circ \overline{cm}_{\overline{loc}} \circ \overline{cn}_{\overline{loc},i}$ with $i \in \{1,...,\lvert\mathbb{CN}\rvert\}$	$\left\{\overline{cs}_{\overline{hs},j}\right\}$ with $j \in \{1,...,\lvert\mathbb{CS}\rvert\}$, $\overline{cs}_{\overline{hs},j} \in \mathbb{CS}$, and $\overline{hs} \in \mathbb{HS}$ (standard compute server with hypervisor)
$\{\overline{vm}_{GPU}\}$ with $\overline{vm}_{GPU} \in \mathbb{VM}$, $\overline{vm}_{GPU}\lvert\mathbb{P}_{HW} \in \mathbb{VM}\lvert\mathbb{P}_{HW}$ (GPU-specialised VM instance without virtual storage)	$\overline{cmp} := \overline{csf} \circ \overline{so}_{\overline{glo}} \circ \overline{cf}_{\overline{glo}} \circ \overline{cm}_{\overline{loc}} \circ \overline{cn}_{GPU,\overline{loc},i}$ with $i \in \{1,...,\lvert\mathbb{CN}\rvert\}$, $\overline{cn}_{GPU,\overline{loc},i} \in \mathbb{CN}$ with $\overline{cn}_{GPU,\overline{loc},i}$ is GPU-specialised compute node instance	$\left\{\overline{cs}_{GPU,\overline{hs},j}\right\}$ with $j \in \{1,...,\lvert\mathbb{CS}\rvert\}$, $\overline{cs}_{GPU,\overline{hs},j} \in \mathbb{CS}$, and $\overline{hs} \in \mathbb{HS}$ (GPU-specialised compute server with hypervisor)
$\{\overline{vs}_{Img}\}$ with $\overline{vs}_{Img} \in \mathbb{VS}$ (virtual storage image instance)	$\overline{cmp} := \overline{csf} \circ \overline{so}_{\overline{glo}} \circ \overline{sf}_{\overline{glo}} \circ \overline{sm}_{\overline{loc}} \circ \overline{is}_{\overline{loc}}$	$\left\{\overline{ds}_{Img,\overline{hs},j}\right\}$ with $j \in \{1,...,\lvert\mathbb{DS}\rvert\}$, $\overline{ds}_{Img,\overline{hs},j} \in \mathbb{DS}$, and $\overline{hs} \in \mathbb{HS}$ (image instance from image respository)
$\{\overline{vs}_{Disk}\}$ with $\overline{vs}_{Disk} \in \mathbb{VS}$, $\overline{vs}_{Disk}\lvert\mathbb{P}_{Arch} \in \mathbb{VS}\lvert\mathbb{P}_{Arch}$ (virtual storage disk instance)	$\overline{cmp} := \overline{csf} \circ \overline{so}_{\overline{glo}} \circ \overline{sf}_{\overline{glo}} \circ \overline{sm}_{\overline{loc}} \circ \overline{bs}_{\overline{loc}}$	$\left\{\overline{ds}_{Block,\overline{hs},j}\right\}$ with $j \in \{1,...,\lvert\mathbb{DS}\rvert\}$, $\overline{ds}_{Block,\overline{hs},j} \in \mathbb{DS}$, and $\overline{hs} \in \mathbb{HS}$ (storage disk instance from block storage device)
$\{\overline{vs}_{Disk}, \overline{vs}'_{Disk}\}$ with $\overline{vs}_{Disk} \in \mathbb{VS}$, $\overline{vs}_{Disk}\lvert\mathbb{P}_{Arch}, \overline{vs}'_{Disk}\lvert\mathbb{P}_{Arch} \in \mathbb{VS}\lvert\mathbb{P}_{Arch}$, $\overline{vs}_{Disk}\lvert\mathbb{P}_{FT}, \overline{vs}'_{Disk}\lvert\mathbb{P}_{FT} \in \mathbb{VS}\lvert\mathbb{P}_{Arch}$, and $\overline{vs}_{Disk} \sim_{Repl} \overline{vs}'_{Disk}$ (virtual storage disk instance with replicated instance)	$\overline{cmp} := \overline{csf} \circ ((\overline{so}_{\overline{glo}} \circ \overline{sf}_{\overline{glo}} \circ \overline{sm}_{\overline{loc}} \circ \overline{bs}_{\overline{loc}}), (\overline{so}_{\overline{glo'}} \circ \overline{sf}_{\overline{glo'}} \circ \overline{sm}_{\overline{loc'}} \circ \overline{bs}_{\overline{loc'}}))$ with $glo' \in \mathbb{GC}$, $loc' \in \mathbb{LC}$ assigned global/local clusters for replication	$\left\{\overline{ds}_{Block,\overline{hs},j}, \overline{ds'}_{Block,\overline{hs'},k}\right\}$ with $j,k \in \{1,...,\lvert\mathbb{DS}\rvert\}$, $\overline{ds}_{Block,\overline{hs},j}, \overline{ds'}_{Block,\overline{hs'},k} \in \mathbb{DS}$, and $\overline{hs}, \overline{hs'} \in \mathbb{HS}$ (storage disk instance with replicated instance at hs')
$\{\overline{vl}(\overline{vm}_1, \overline{vm}_2)\}$ with $\overline{vl} \in \mathbb{VL}$, $\overline{vm}_1, \overline{vm}_2 \in \mathbb{VM}$ (virtual link instance between two VM instances)	$\overline{cmp} := \overline{csf} \circ \overline{so}_{\overline{glo}} \circ \overline{nf}_{\overline{glo}} \circ \overline{nm}_{\overline{loc}} \circ \overline{nn}_{\overline{loc},i}$ with $i \in \{1,...,\lvert\mathbb{NN}\rvert\}$	$\left\{\overline{con}_{\overline{hs},j}(\overline{cmp}(\{\overline{vm}_1\}), \overline{cmp}(\{\overline{vm}_2\}))\right\}$ with $j \in \{1,...,\lvert\mathbb{CON}\rvert\}$, $\overline{con}_i \in \mathbb{CON}$, and $\overline{hs} \in \mathbb{HS}$ (end-to-end connection between assigned compute server, usually virtual overlay between VM instances)
$\{\overline{vl}(\overline{vns}_{Gate}, \overline{vns}_{Rout})\}$ with $\overline{vl} \in \mathbb{VL}$, $\overline{vl}\lvert\mathbb{P}_{Access} \in \mathbb{VL}\lvert\mathbb{P}_{Access}$, $\overline{vns}_{Gate}, \overline{vns}_{Rout} \in \mathbb{VNS}$ (virtual link instance providing access to the network via gateway and routing service)	$\overline{cmp} := \overline{csf} \circ \overline{so}_{\overline{glo}} \circ \overline{nf}_{\overline{glo}} \circ \overline{nm}_{\overline{loc}} \circ (\overline{nn}_{\overline{loc},Gate,h}, \overline{nn}_{\overline{loc},Rout,i})$ with $h,i \in \{1,...,\lvert\mathbb{NN}\rvert\}$, and $\overline{nn}_{\overline{loc},Gate,h}$ offers gateway service and $\overline{nn}_{\overline{loc},Rout,i}$ routing service	$\left\{\overline{con}_{\overline{hs},j}(\overline{cis}_{Gate,\overline{hs}}, \overline{cis}_{Rout,\overline{hs}})\right\}$ with $j \in \{1,...,\lvert\mathbb{CON}\rvert\}$, $\overline{con}_i \in \mathbb{CON}$, and $\overline{hs} \in \mathbb{HS}$ (access network with communication service instances for gateway $\overline{cis}_{Gate,\overline{hs}}$ and routing $\overline{cis}_{Rout,\overline{hs}}$)
$\{\overline{vns}_{DHCP}\}$ with $\overline{vns}_{DHCP} \in \mathbb{VNS}$ (virtual network service instance providing DHCP)	$\overline{cmp} := \overline{csf} \circ \overline{so}_{\overline{glo}} \circ \overline{nf}_{\overline{glo}} \circ \overline{nm}_{\overline{loc}} \circ \overline{nn}_{\overline{loc}}, i$ with $i \in \{1,...,\lvert\mathbb{NN}\rvert\}$	$\left\{\overline{cis}_{DHCP,\overline{hs},j}\right\}$ with $j \in \{1,...,\lvert\mathbb{CIS}\rvert\}$, $\overline{cis}_{DHCP,\overline{hs},j} \in \mathbb{CIS}$, and $\overline{hs} \in \mathbb{HS}$ (DHCP communication service instance)

[a] With $\overline{csf} \in \mathbb{CSF}, \overline{so}_{glo} \in \mathbb{SO}, \overline{cf}_{glo} \in \mathbb{CF}, \overline{cm}_{loc} \in \mathbb{CM}, \overline{cn}_{loc} \in \mathbb{CN}, \overline{sf}_{glo} \in \mathbb{SF}, \overline{sm}_{loc} \in \mathbb{SM}, \overline{bs}_{loc} \in \mathbb{BS}, \overline{nf}_{glo} \in \mathbb{NF}, \overline{nm}_{loc} \in \mathbb{NM}, \overline{nn}_{loc} \in \mathbb{NN}$, and $glo \in \mathbb{GC}, loc \in \mathbb{LC}$ assigned global/local clusters

practices on data centre design and cloud architectures were investigated. This provides a good basis for a plausible and self-contained model. Since the notation of an ontology is used for the construction of the model, it is even possible to perform, in a next step, a formal verification of the characteristics described. The formal verification of the model is outside the scope of this thesis and not further investigated. In Section 7.3, an outlook on how to perform such a formal verification is given.

Having understood the structure of IaaS cloud infrastructures, it is now possible to investigate the technical capabilities on security and compliance management in such clouds. Section 4.2 investigates the cloud security management and Section 4.3 analyses compliance management in clouds.

4.2 Cloud security management

Cloud security is a widely investigated research area. There are several surveys and guidelines identifying security and privacy challenges within cloud computing (e.g., [117] [191] [41] [230]). Figure 4.14 provides a comprehensive (but not necessarily exhaustive) overview of the security and privacy challenges mentioned in the literature according to the classification scheme provided by NIST [117].

The research challenges addressed in this thesis (cf. Section 1.3) fall into the area of *compliance* and in particular into the area of *law and regulations* and of *data location*. More specifically, the technical challenges in cloud computing of implementing the legal requirements identified in Section 3.6 are addressed. Additionally, *visibility* in the context of compliance monitoring and *auditability* in the context of providing evidence in compliance reports are addressed.

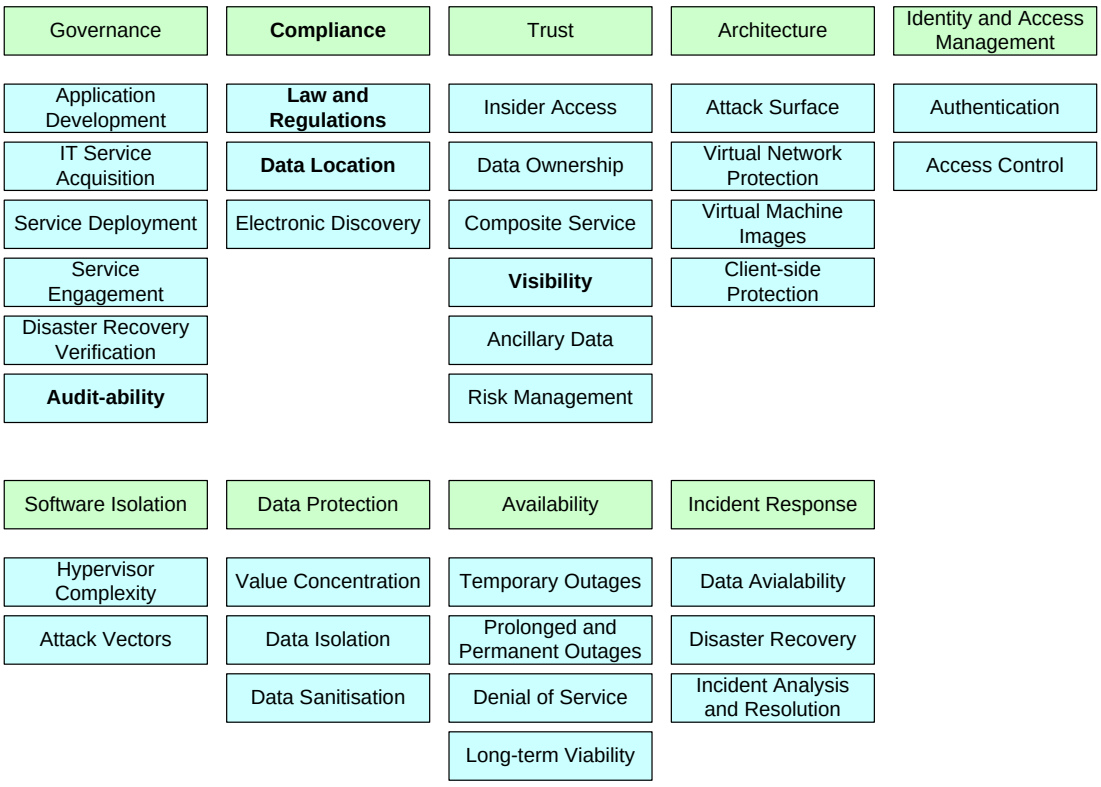
In the following, the capabilities and technical prerequisites required to comply with the identified legal requirements are analysed in general and with respect to existing literature and current practice. This section covers the analysis of capabilities and technical prerequisites of cloud security management with respect to the legal requirements identified in the legal analysis (cf. Section 3.6). The capabilities and prerequisites for documenting, monitoring, and reporting to support cloud compliance management are examined in Section 4.3.

4.2.1 Effective level of security

According the Def. 3.2, the *effective level of security* specifies the implemented security measures at the location of data processing (i.e., at the hosting site). The cloud provider has to ensure that the cloud management process assigns virtual resources only to hosting sites, which have an *effective level of security* that satisfies the *necessary level of security* applicable to hosted virtual resources.

This refers to the location inhomogeneity identified in cloud infrastructures (cf. Definition 2.2). Location homogeneity implies that both *ensured legal framework conditions* and *effective level of security* are the same for all hosting sites. Location inhomogeneity implies that for some hosting sites the *ensured legal framework conditions* or the *effective level of security* are not the same. If the *ensured legal framework conditions* are not the same then also the

Figure 4.14: Classification security and privacy challenges in cloud computing according to NIST [117] and other existing surveys and guidelines [191] [41] [230].



ensured level of security is not the same either, since the obligations when it comes to security measures differs at those locations with different legal framework conditions. In any case, in legally compliant clouds, the *effective level of security* can be higher than the *ensured level of security* but legal framework conditions may bypass implemented security measures anyway since they are not ensured For instance, extensive access privileges by investigative authorities in the USA render access and transfer control measures useless in general. Further, the *ensured level of security* specifies the minimum level of security which has to be implemented at a specific location. Assuming legal compliance (i.e., the *effective level of security* is never lower than *ensured level of security*),[1] the cloud provider has to verify that each hosting site which is involved in data processing has an *ensured level of security* satisfying the *necessary level of security* for the data being processed. In the following, the challenge of deciding on the *ensured level of security* and of enforcing the *necessary level of security* is defined as the *challenge of location inhomogeneity*:

Definition 4.6 (Challenge of location inhomogeneity) *The **challenge of location inhomogeneity** the need to describe the context of determin and enforcement within an information model and for methods to identify, to decide based on* ensured level of security *(which is minimum for* effective level of security*), and to enforce the* necessary level of security*.*

*The **information model** has to be able to express the applicable requirements with respect to (1) the location of the data processing, (2) the category of the processed data, (3) the origin of the processor, controller, and data subjects or 'data owner', and (4) the applicable requirements from contracts and SLA (cf. Section 3.6.1). For proper decision and enforcement, the **identification method** has to be able to acquire reliabe information on the data processing location and its* ensured level of security*. Based on the information model and the information provided by the identification method, the **decision and enforcement methods** have to ensure that the resource allocation of the cloud management process complies with the* necessary level of security *(i.e., the* ensured level of security *at the location of allocated hardware resources satisfies the* necessary level of security *that is required for the requested virtual resources).*

In the literature, the challenge of location inhomogeneity is addressed in the context of information privacy with the terms 'data location' [117][41], 'data locality' [191], and 'multi location issue' [230], which are allocated in the intersection of, on the one hand, location-centric and -aware data processing, and on the other hand, information- and data-centric security.

In the area of location-centric and -aware data processing, there is a good deal of research on user location and privacy in mobile cloud computing [71] and in location- and context-aware systems [16], but none of this research addresses the location of data processing and the *necessary level of security* within the cloud.

The research area of information- and data-centric security seeks to disentangle the security from the underlying communication, storage and processing mechanisms and applying security mechanisms directly to data. Examples of this are the XML-based security standards XML-Encryption[60] and XML-Signature [18], or security mechanisms for identifier integrity and

[1] In any case, the cloud provider regularly has to ensure the compliance of subcontractors, e.g., by contractual agreements including inspections of compliance and the effectiveness of any security measures implemented.

origin verification used in information-centric networking [2] [84]. Information- and data-centric security mechanisms are designed to apply security to data directly and independently of the processing environment. Therefore, information- and data-centric security mechanisms are able to attach security information (e.g., security policies) to data and make them available for data processing systems, but (intentionally by design) it does not take into account the location of data processing and the *necessary level of security*.

Methods for documenting and retrieving the origin of data are investigated in the area of data provenance, for example, using documentation and annotation of data processing and data transfer[145] and modelling data provenance on the World Wide Web (WWW) [229] enabling queries on the origin of the data. These methods can be used to identify the creator of specific data, but the location of the data and its processing as well as the *necessary level of security* are not considered.

In general, the challenge of location inhomogeneity can be considered to be related to the area of information flow analysis, since it requires an information model used for identification, decision on, and enforcement of information flows. Particularly related to the challenges are the investigations on modelling access and mobility control using network references [87], which makes it possible to describe end-to-end security policies [201]. Further, methods of information flow analysis can be applied to detect co-location of virtual machines based on hypervisor information and used to protect from side-channel attacks through VM migration and halting/resuming strategies [14]. To conclude, none of these research works directly addresses the challenges of the *necessary level of security*. Nonetheless, information flow analysis provides methods to model and verify security requirements in the context of distributed systems, which make the methods of information flow analysis a good candidate to describe the information model covering determination and enforcement of the *necessary level of security*. In particular, it has been observed that information flow control policies can be used to model legal regulations [14]

In practice, the challenge of location inhomogeneity is addressed by solutions providing nationally hosted cloud computing (e.g., Initiative Cloud Services Made in Germany,[1] AWS GovCloud (US),[2] FUJITSU Cloud IaaS Private Hosted[3]). To the best of the author's knowledge, there is only one approach addressing technically the validation of the *effective level of security* of hosting sites in cross-boarder cloud computing. There also exists an approach that partially addresses the challenge of location inhomogeneity by reliably identifying the location of virtual machines. Both approaches are discussed below.

The EU FP7 integrated project OPTIMIS[4] addresses the *necessary level of security* with respect to European data protection law. In the approach [202], the *necessary level of secu-*

[1]The initiative called Cloud Services Made in Germany provides a platform on which to promote cloud provider located in Germany and explicitly governed by German legislation. On the Internet (in German): http://www.cloud-services-made-in-germany.de/ (last visited: 30.06.2015)

[2]AWS GovCloud (US). On the Internet: http://aws.amazon.com/govcloud-us/ (last visited: 30.06.2015)

[3]FUJITSU Cloud IaaS Private Hosted addresses the challenge of "geographically-specific regulations" and is "tailored to the specific needs of the local territory". On the Internet: http://www.fujitsu.com/global/services/infrastructure/iaas/ (last visited: 30.06.2015)

[4]EU FP7 integrated project OPTIMIS. On the Internet: http://www.optimis-project.eu/ (last visited: 30.06.2015)

rity is described using WS-Agreement (cf. WS-Agreement standard [9]) by listing all counties and hosting sites that satisfy the *necessary level of security* [231]. Decision and enforcement on these listings are performed during resource placement. The location of virtual machine and virtual storage placement is identified via IP address range verification and exploitation of the rack awareness features of Hadoop Distributed File System (HDFS) (cf. the Hadoop user guide [10] and the description of rack awareness [123]). The decision is based on whether a hosting site is located within the European Union or not. The enforcement is done before resource allocation and continuously monitored during operation to consider resource migration. Addressing the problem for virtual machine and virtual storage placement, the reliability of the verification depends on the correct allocation of rack identifier and IP address ranges, which are both under control of the hosting sites and have no implemented security or integrity mechanisms. They are particular easy to manipulate. The IP address of a network connection endpoint can be unrecognisably disguised by using network address translation. The rack identifier used for the rack awareness depends on the association with its parents node, which is "proprietary to each organization" [123]. Thus, the rack identifier can be defined arbitrarily by the hosting site, and consecutively, abusable to obfuscate the physical location, for example, through linking them with parent nodes of other hosting sites. The approach of OPTIMIS addresses the validation of the *ensured level of security*, but does not provide generally applicable and trustworthy mechanism[1] to identify the location of virtual resources and their assigned hardware resources. However, it is generally possible to locate data in the cloud in a trustworthy manner in combination using geo-location methods and proof of storage (POS) protocols [3] [4], neither of which is applied in OPTIMIS. Further, the approach uses a policy language to describe requirements for the *necessary level of security*, but do not provide an information model to describe locations and requirements in general. However, the approach is a proof of concept for considering the *necessary level of security* in cloud infrastructures.

The EU FP7 integrated project, TCloud,[2] investigates – among other topics – privacy and data protection by identifying the location of virtual machines related to the hosting hypervisor and compute server using trusted hypervisors and trusted platform modules in OpenStack [180, pp. 12 et seqq.]. Therefore, an information model is used to describe the location of virtual machines dependent on the hosting compute server [1]. This information model allows the decision on and enforcement of the location of virtual machines based on co-location and previously assigned context-information (e.g., integrity level and location). While providing a reliable method to identify the hosting compute server of a virtual machine, the approach addresses neither the *necessary level of security* nor the *effective level of security*. In particular, no methods to for automated decision making and enforcement are provided. However, it is possible that the cloud customer manually assigns constraints to requested virtual machines specifying preferred compute servers [53, pp. 55 et seqq.]. Such a manual assignment requires knowledge of the physical infrastructure of the cloud that a cloud customer does not usually have, and moreover, manual assignment lacks the support of scalability and rapid elasticity

[1] The approach depends on the rack awareness feature of HDFS running "on a cluster of computers with a tree hierarchical network topology" [123] and its association with IP address ranges, neither of which is either general or secure.

[2] EU FP7 integrated project TCloud, on the Internet: http://www.tclouds-project.eu/ (last visited: 30.06.2015).

required in cloud computing (cf. Section 2.1.1). To conclude, the approach of TClouds does not solve the challenge of location inhomogeneity but provides a proof of concept on (1) securely identifying the compute server that is hosting a specific virtual machine by using trusted computing and (2) enforcing the assignment of virtual machines to a specific compute server in OpenStack.

Further, there exists a draft of a proof of concept implementation of trusted geo-location in clouds [217] showing that it is possible to geo-tagging cloud resources and enforce resource allocation by using trusted computing pools in OpenStack. While this approach addresses the need for geo-location-aware resource allocation in clouds, the approach lacks an information model describing all location constraints of corporate customers and ensuring a correct (i.e., free of conflicts) decision making and enforcement in respect to location constraints, when allocating cloud resources. However, the implementation uses a similar (but different) approach as presented in the proof of concept implementation in this thesis. Therefore, similarities and differences are discussed in more detail in Section 6.1.2.

In this thesis, the challenge of location inhomogeneity is addressed (1) by defining an information model using methods of information flow control in Section 5.3, (2) by identifying reliable methods to identify the location of data processing in cloud computing in Section 5.4.1, and (3) by identifying reliable and scalable methods of decision making and enforcement and applying them to the cloud management process in Section 5.4.2.

4.2.2 Cloud security policies

The basis for deciding on the *ensured level of security* and enforcing the *necessary level of security* within the cloud management process is the information model expressing the security requirements of cloud customers and cloud provider. These requirements are expressed and communicated via security polices, which are on the one hand the organisational standard and on the other hand the technical representation of all applicable security requirements. Thus, security policies enable the cloud provider to communicate security requirements to the hosting site and to implement automated decision making and enforcement within the cloud management process. According to Section 3.6.2 the following requirements have to be satisfied by any security policy.

- It is expressed in a **technically enforceable form** to be processable by the cloud management process as well as by the configuration and security management functions at the hosting sites.
- It is possible to **express all applicable rules**, including the mapping of data types with applicable safeguards. In particular, the policies language **is extendible** allowing to describe new requirements that may arise in future.
- To **support multi-tenancy**, rules are expressible and distinguishable by cloud customers.
- It is possible to **identify and cope with conflicts** between different sets of rules. In particular, there are defined merge and intersection operators for combining different sets of rules.

- It is **legible and comprehensible** to enable review and audit processes that may apply before, during, and after data processing.

The literature identifies several approaches to express legal and security requirements using security policies. An important area is the modelling of legal regulations for software and system engineering [158] [159]. This area covers several methods, for example, methods of symbolic logic, first-order temporal logic, markup-based representations, and goal modelling. It is a general observation in this area that modelling legal regulations directly often struggles with the complexity and ambiguity of legal regulations [158]. Comparing the existing methods for modelling legal requirements, markup-based representations and in particular Extensible Markup Language (XML) [27] turns out to be the most suitable approach for expressing legal requirements in a technically enforceable and legible form [158]. Additionally, XML is an expressive and flexible language that is fully specifiable and verifiable using language schemes like XML Schema Definition (XSD) [199] and relax NG [45]. This makes XML a good candidate to specify security policies for the automated decision making and enforcement within the cloud management process.

There already exist several XML-based policy languages, for example, in the context of access control [223], privacy management [125], and trust management [182]. Further, many of the existing XML-based security standards are applicable in the context of cloud security [186]. In the context of cloud computing, there are also approaches to express SLA for example by defining a novel language [163] or extending the WS-Agreement standard[1] [231]. However, none of these XML-based policy languages express the required information for enforcing the *necessary level of security* (i.e., location, data type, origin, and applicable requirements; cf. Def. 4.6). In any case, an extension of an existing policy language or the definition of a novel policy language is required to express these requirements.

In practice, there is little information on the security policies used in existing commercial cloud infrastructures and their expressiveness, since they are dealt with in the context of the internal, non-transparent cloud management process.

The investigation of open cloud infrastructures reveals that there is only little standardisation and little support of security requirements. For example, OpenStack[2] uses the standard on JavaScript Object Notation (JSON)[3] to define key-value pairs, which are specific to OpenStack. In OpenNebula,[4] predefined lists of key-value pairs specific to OpenNebular are used. Here, the focus lies on load balancing, and security requirements are not considered. Eucalyptus uses predefined policies for resource scheduling, which supports throughput, response time, and fairness / waiting time [192]. There is no support for security requirements, but it is possible to implement new schedule algorithms [192]. The approach of the EU FP7 integrated project OPTIMIS extends the WS-Agreement standard[5] by introducing white-listing for countries and hosting sites with an *adequate level of protection*.

[1] WS-Agreement standard [9].
[2] OpenStack documentation on policies, on the Internet: `http://docs.openstack.org/developer/glance/policies.html` (last visited: 30.06.2015).
[3] The JSON Data Interchange Format [66].
[4] OpenNebula documentation on resource scheduling, on the Internet: `http://archives.opennebula.org/documentation:rel4.4:schg` (last visited: 30.06.2015).
[5] WS-Agreement standard [9].

In this thesis, a security policy is required to perform an experimental evaluation implemented in OpenStack (cf. Section 6.1). However, the identification and development of a suitable security policy is not the scope of this thesis. Therefore, a simplistic XML-based policy language is defined to express the required information for deciding on and *ensured level of security* and enforce the *necessary level of security* according to the information model described in Section 5.3 and make them available to the cloud management process in OpenStack. Possible approaches to the development of more sophisticated security policies are discussed in Section 7.3.

4.2.3 Security measures in the cloud

Applicable legislation and the cloud customer's preferences define the safeguards that have to be implemented. These safeguards aim to protect the data processing on behalf of the cloud customer and support the enforcement of requirements described in SLA.

There are four areas identified by the legal analysis in Section 3.6.3 that have to be covered by implemented and enforced safeguards within the cloud: (1) basic security measures, (2) access control, (3) transfer control, and (4) countermeasures and incident response. All of these areas are analysed in the following sections.

4.2.3.1 Basic security measures

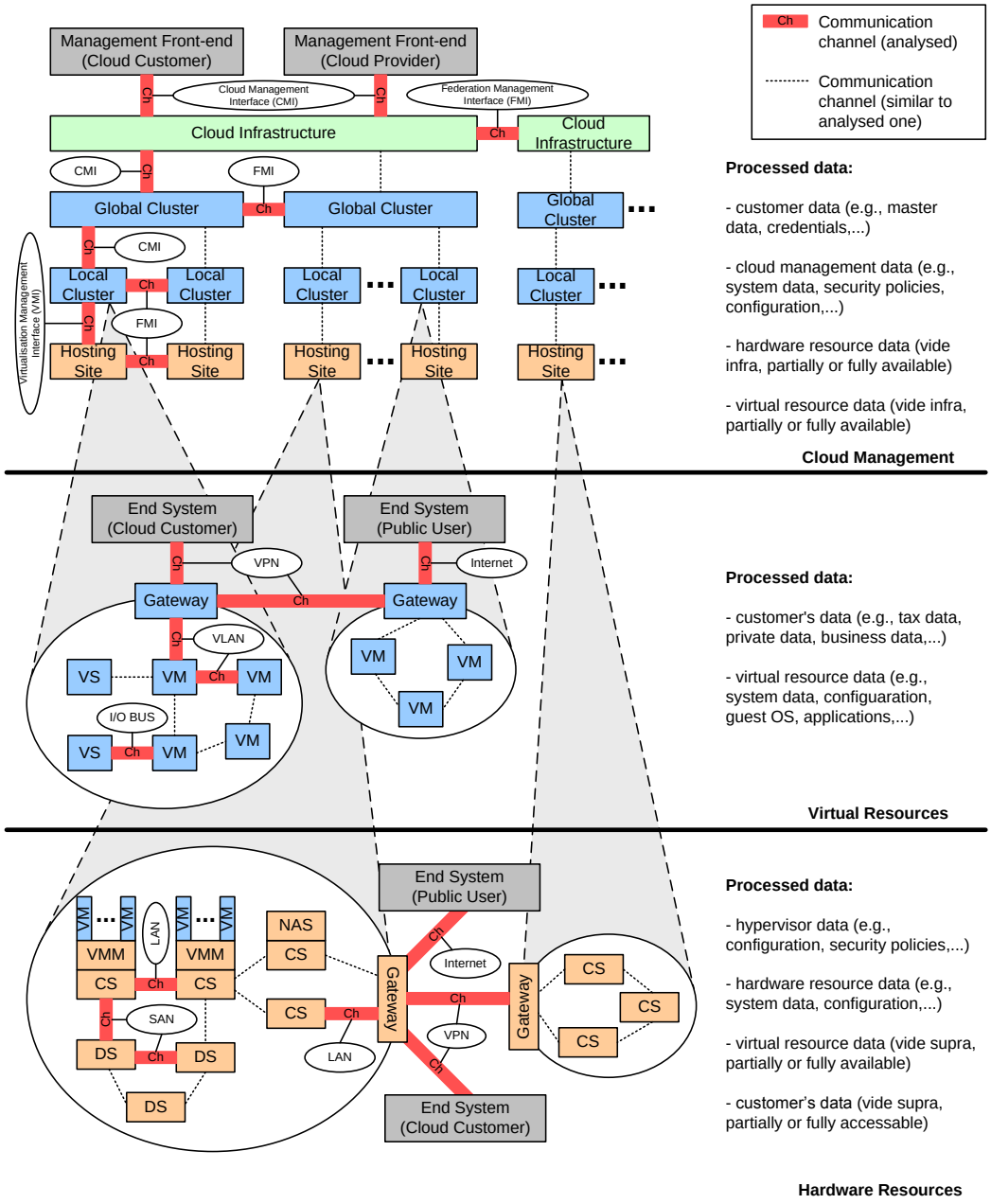
Basic security measures aim to secure communication and secure data storing/processing. This is achieved by applying methods for ensuring the confidentiality, integrity, and availability of virtual resources, hardware resources, and the cloud management process, including data stored and processed within. Additionally, these methods support the effective operation of mechanisms for access control (cf. Section 4.2.3.2) and transfer control (cf. Section 4.2.3.3).

Communication and data processing in the cloud In cloud infrastructures, basic security measures can be applied to different levels of resource abstraction and management, i.e., (a) cloud management process, (b) virtual resources, and (c) hardware resources. Each of these levels has specific communication relations and processed types of data that are depicted in Figure 4.15.

On the **cloud management level** (cf. Figure 4.15), there are three different types of communication relation: (1) the vertical communication of the cloud management using Cloud Management Interfaces (CMIs) (e.g., OCCI [142] and CIMI [54]) to manage and operate virtual resources), (2) the horizontal communication of the federation management using Federation Management Interfaces (FMIs) (e.g., cross-cloud federation manager [38] and virtual execution environment manager [171]), and (3) the communication between the local clusters and the hosting sites using Virtualisation Management Interfaces (VMIs) (e.g., Open Virtualization Format (OVF) [56] and Virtualization Management (VMAN) [55]).

The processed data at this level are customer data (e.g., master data to manage the contract with the customer, and credentials for connecting with the cloud infrastructure) and cloud management data necessary for operating the cloud infrastructure (e.g., system data of the

Figure 4.15: Overview of communication relations and processed data in IaaS clouds



infrastructure, applied security policies, and configuration data of managed systems and provided services). In particular, configuration data of virtual resources and hardware resources are processed for management purposes.

On the **virtual resource level** (cf. Figure 4.15), there are four different types of communication relation: (1) VPN used to establish private access networks (e.g., IPsec [119] and

SSL/TLS [205]), (2) VLAN for connecting virtual machines (e.g., IEEE 802.1Q [106] and VPLS [121][130]), (3) Internet-based application transport protocols (e.g., HTTP for WWW-based applications), and (4) (virtual) I/O busses used to connect virtual storage with virtual machines (implemented by using, for example, HDFS [10] for block storage and NFS [184] for object storage).

Processed data on this level are cloud customer's data that are transmitted from end systems to virtual resources for data processing within the cloud. Cloud customer's data can cover, for example, tax data, private data, and business data (cf. identified data types in Section 3.5.3.3). Additionally, virtual resource data are processed, including the system and configuration data of the virtual resources and the data on the hosted guest OS and applications.

On the **hardware resource level** (cf. Figure 4.15), there are four different types of communication relation: (1) LANs providing switched Ethernet (possibly in conjunction with VLAN technologies [44, pp. 2–1 et seq.]) for connecting the compute system (CS) hosting virtual machines and other applications, like NAS, (2) SANs providing switched Ethernet (possibly in conjunction with VLAN technologies [44, p. 2–2]) specialised for storage provisioning, (3) VPNs implementing the private access networks of the virtual resource layer (vide supra), and (4) Internet-based transport protocols implementing the public access network (vide supra), respectively.

Processed data on this level are the hypervisor data, including applied configurations and security policies, and the hardware resource data, for example, applied configurations and system data of the compute server and data storage units. Additionally, the virtual resource data (vide supra) are fully or partly available through the management functions of the hypervisors that are also used by the VMI. Moreover, the customer's data are fully or partially accessible using virtual machine introspection [149] or via direct hardware access [178].

Communication security The **integrity and confidentiality of communication channels** in IP-based networks is well understood and can be established in many ways, for example:

- by using secure transport protocols addressing confidentiality and integrity, e.g., Hypertext Transfer Protocol Secure (HTTPS) [42] instead of Hypertext Transfer Protocol (HTTP) communications (e.g., used in OpenNebula for communication with the OCCI server [156]);
- by using secure VPN and VLAN protocols, e.g., IPsec (supports confidentiality [119]) and SSL/TLS (supports confidentiality and integrity [205]); and
- by using XML security standards (e.g., XML encryption [60] and XML signature [18]) on XML-based message formats (for example, applicable to cross-cloud federation manager [38] and VMAN [55]).

In particular, there exist approaches on communication security that are specific to cloud management, for example, Cloud Data Management Interface (CDMI) implements transport security for cloud storage management [190, p. 33]).

The **availability of communication channels** in IP-based networks is also well understood. There exist accurate models for measurement [213]. There are multiple identified

threats to availability, for example, hardware failure and congestion due to extensive use or attacks [181]. Further, there exist multiple approaches addressing these threats, for example, implementing diversity for improved availability [183] and introducing intrusion detection and prevention [36] to mitigate attacks on availability. In practice, the availability of communication channels in virtualised infrastructures can be implemented by redundant hardware resources [44, pp. 2–12 et seqq.].

To conclude commutation security in cloud infrastructures, there are methods implementing confidentiality, integrity and availability on communication channels used in cloud infrastructures. In particular, the communication between the cloud infrastructure and the management interfaces as well as the communication between cloud infrastructures can be protected.

Data security In general, the **confidentiality of data** is addressed by using cryptosystems for data encryption (e.g., AES [151] and DES [150]). It is possible to use such cryptosystems to encrypt data storage [96] and memory [224] during system operation. Such approaches protect from unauthorised attempts to read the data directly from the data storage or memory (e.g., via direct hardware access [178]). However in virtual machines, these techniques are less effective against attempts to use virtual machine introspection [149], since the data are decrypted before it is processed in virtual machines. A possible solution for this issue is to use homomorphic cryptoschemes, which allow processing to be performed on encrypted data [147]. Alternatively, the usage of trusted hypervisors can help to prevent unauthorised virtual machine introspection [79]. In the context of data privacy, there exist approaches using data aggregation [48] and transformation [198] to remove private data ensuring a certain degree of anonymity. These methods can help to ensure the confidentiality of data by removing the protected data or reducing the granularity of information in the data. Therefore, they are not suitable for lossless data processing.

The integrity of data is usually addressed by detection and correction methods to address possible modification of the processed data (e.g., cryptographic hashes [59] and error-correcting codes (ECCs) [94]). Using cryptographic hashes combined with signature schemes additionally provides guarantees on the detection of modifications, since the detection code is protected, too (e.g., DSA [152]). There also exist legal standards and regulations on the use of signature schemes for electronic signatures (e.g., the German signature law and the Electronic Signature Directive 1999/93/EC).

The availability of data aims to guarantee access to data and protect data from loss. There are several strategies for guaranteeing access, including storage redundancy in disk arrays [77], data duplication [138] and globally distributed storage replication [124]. There exist best practices on storage high-availability for hosting sites [44, pp. 2–17 et seq.], and there exist cloud services offering increased data availability (e.g., geo-replication in Windows Azure Storage [143]). To protect from data loss, there exist several approaches and implementations on backup and recovery strategies [40], which can be applied to virtual storage as well [138]. There also exist backup and recovery services for virtual storage in cloud infrastructures (e.g., the snapshot function of Amazon S3 [5] and Fujitsu Backup as a Service [75]).

To conclude data security in cloud infrastructures, there exist solutions implementing

confidentiality, integrity, and availability on a hardware resource and virtual resource level. Additionally, these solutions can be applied to data processing in cloud management (e.g., using database encryption [50] in combination with storage availably and signature schemes to protect customer data). However, basic security measures in virtual resources may render ineffective when the underlying hypervisor is exploited to attack the hosted virtual machines (e.g., by using virtual machine introspection [80]). Here, virtualisation security is required [78], which is operated and managed by the cloud provider [43]. However, virtualisation security is not recognisable on the level of virtual resources (but only on hypervisor level). Without the cooperation of the cloud provider, the cloud customer has no opportunity to gain any information on implemented virtualisation security in the cloud.

Conclusion and relevance in this thesis The analysis of basic security measures shows that there already exist applicable solutions for secure communication and data processing in cloud computing. A particular issue is the possibility of the hypervisor undermining basic security measures on the virtual resource level. This issue is addressable by trusted hypervisors. Such security measures are not visible to the cloud customer and reporting by the cloud provider is required to provide evidence of such measures being applied to the cloud customer. In this thesis it is assumed that basic security measures are applied to ensure a secure operation of the cloud infrastructure. In the following, the implementation and establishment of basic security measures for communication and data security in cloud infrastructures are not further investigated. Methods for monitoring and reporting on the effective operation of basic security measures are discussed in Section 4.3.

4.2.3.2 Identity and access management

According to the legal analysis (cf. Section 3.6.3.2), it is necessary to implement access control mechanisms to ensure the authentication and authorisation of data processing entities (e.g., cloud providers, cloud customers, and hosting sites).

This requires the implementation of identity and access management on the cloud providers site as well as on the cloud customers' sites and at hosting sites. The identity management has to ensure the authenticity of actors (i.e., employees and entitled entities of cloud providers, of the cloud customers, and of the hosting sites) that are involved in data processing and accessing cloud infrastructures and provided resources. Based on the identity and authorisation of the accessing actors, the access management has to enforce access control on virtual resources, hardware resources, and the cloud management process, including data stored and processed within these. To provide evidence on enforcement of access control, access management has to document for each access attempt (using logging mechanisms) whether access was granted or not. Such evidence also supports the non-repudiation of actions (e.g., data access and service operation) performed by the accessing actors.

In the literature, identity and access management is investigated in many directions including the context of authentication methods and protocols (e.g., Kerberos [114]), standards and protocols for exchanging authentication and authorisation data (e.g., SAML [35] and OAuth [95]), identity management systems (e.g., Liberty Alliance [34] or Shibboleth [37]),

access control models [176] (particularly role-based access control (RBAC) [177] and privacy-aware RBAC [153]), and use-case-specific approaches (e.g., user-centric management [61] and federation management [76]).

In particular, identity and access management is mentioned in standards on cloud management interfaces, for example CDMI designed for cloud storage management supports authentication, authorisation and access control [190, p. 33], and the OCCI standard recommends to use Transport Layer Security (TLS) for authentication [142, p. 22]

In practice, many cloud infrastructures support identity and access management. For example, the OpenStack Identity Service Keystone implements an identity management system providing authentication and authorisation mechanisms for accessing virtual resources, hardware resources and components of the cloud management process (cf. 'Keystone' in OpenStack documentation [157]). Additionally, there exist extensions for OpenStack, introducing OpenID authentication [120] and SAML authentication [49]. Other examples are the authentication mechanisms of OpenNebula (cf. 'User Security and Authentication' in OpenNebula documentation [156]) for Amazon S3 (cf. 'Access control' and 'Signing and Authenticating REST Requests' in Amazon S3 documentation [5]) and of Windows Azure (cf. 'Multi-Factor Authentication' in Windows Azure documentation [143]).

To conclude, there exist several solutions applicable to identity and access management. In particular, authentication and authorisation methods and protocols are well understood. Access control is often modelled using a role-based approach. Existing cloud infrastructures usually implement identity and access management mechanisms for virtual resources, hardware resources, and the cloud management process.

In this thesis, identity information on location and implemented security measures of virtual resources, hardware resources, and hosting sites is used to identify the *effective level of security*. Such types of information are not usually provided by identity management systems in cloud infrastructures, since they are not required for identification and authorisation. It is assumed that an identity and access management system is used to manage access control on virtual resources, hardware resources, hosting sites and components of the cloud management, which is in the case in OpenStack (cf. 'Keystone' in OpenStack documentation [157]). Further, it is assumed that it is possible to extend the information basis of the identity and access management system through informations on location and implemented security measures, which is possible in principle as shown in this thesis for OpenStack (cf. Section 6.1). The relevance of identity and access management for this thesis is limited to obtaining and managing information on location and implemented security measures of virtual resources, hardware resources, and hosting sites. Thus, other methods for authentication and authorisation in cloud computing are not further investigated.

4.2.3.3 Transfer control

According to the legal analysis (cf. Section 3.6.3.3), the cloud provider has to implement transfer control mechanisms to ensure (1) the authenticity and authorisation of recipients, (2) the satisfaction of the *necessary level of security* at the recipient's location, and (3) compliance with the applicable transfer restrictions. Further, transfer control has to consider data transfer to other cloud customers, subcontracted hosting sites and third party service providers, third

parties accessing virtual resources of the cloud customer (e.g., cloud user or supervisory authorities), and in the case of cloud federation, third party cloud providers (cf. Section 3.6.3.3). Consequently, transfer control has to be applied on any type of data transfer. This includes data transmission via communication channels on the one hand, and creation and migration of virtual resources used for processing data on the other hand.

Figure 4.16: Data transfer in IaaS cloud infrastructures from the cloud provider's perspective

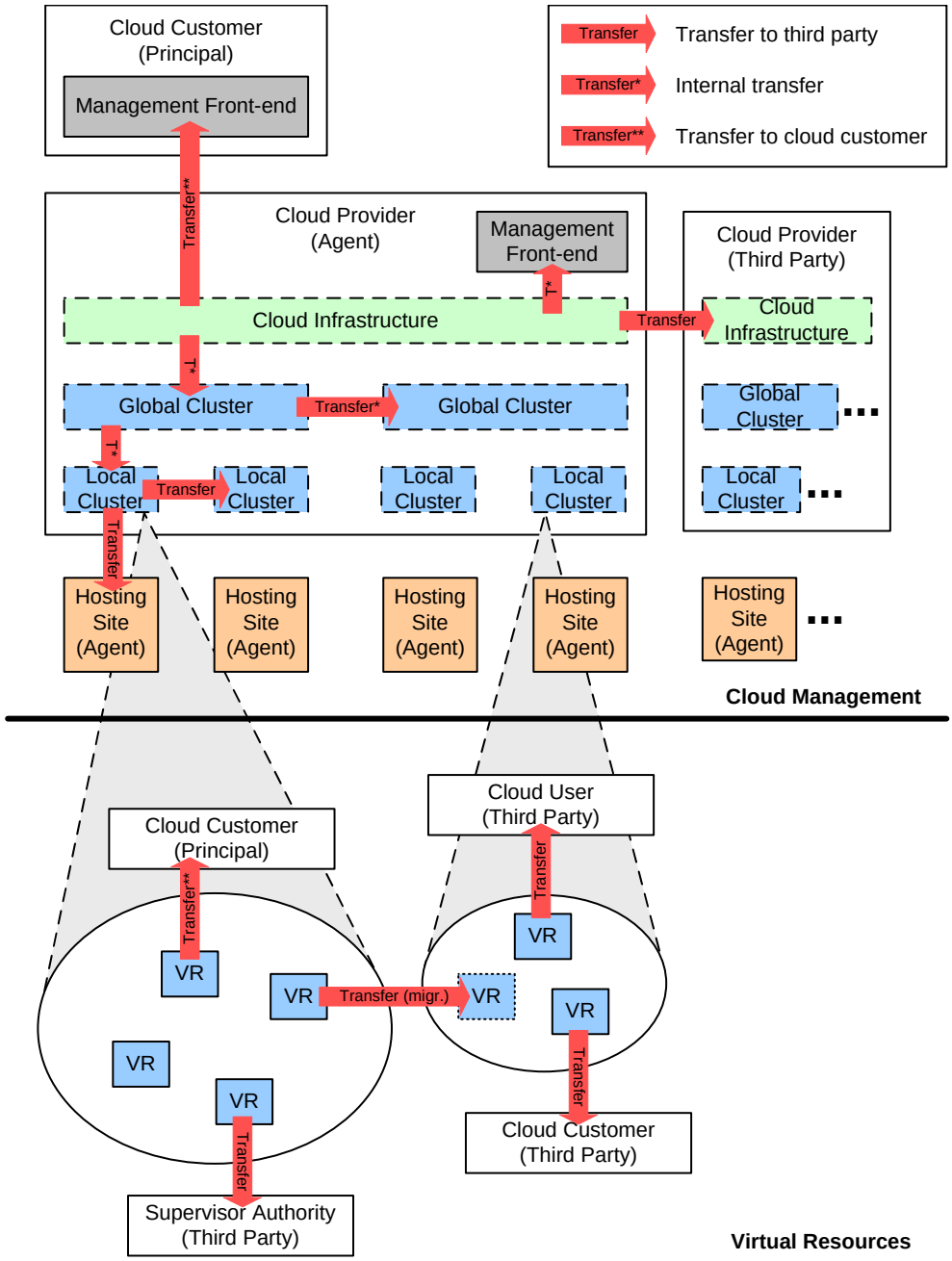


Figure 4.16 depicts the possible data transfers in IaaS cloud infrastructures as described above. The possible data transfers can be classified by (a) transfer to third party (i.e., data transfer in the legal sense; cf. Section 3.2), (b) transfer internal of the cloud provider (depending on the location of hosting sites operating the cloud management infrastructure this can result in a cross-boarder transfer, and therefore in data transfer in the legal sense; cf. Section 3.2), and (c) data transfer to the cloud customer (depending on the location of the cloud customer this also can result in a cross-boarder transfer). In the following, methods for implementing transfer control are discussed.

The **authenticity and authorisation of recipients** can be ensured by using methods of identity and access management based on the recipients' identity (cf. Section 100). The identity and access privileges of recipients can be managed analogous to those used for access control, and the methods used for verifying the authenticity and authorisation of accessing identities apply analogously for recipients. For example, a role-based access control model can be applied to describe and verify the recipients' identities and their authorisation (e.g., for privacy-sensitive data [153]).

The **satisfaction of the *necessary level of security*** at the recipients' location is a mandatory prerequisite for legally compliant data transfer (cf. Section 3.6.1). Therefore, the *ensured level of security* at the recipient's location has to be verified, and according to Section 4.2.1, the location of the recipient, the transferred data types, the sender's origin, and applicable transfer restrictions are used for the decision on the *ensured level of security* and enforcement of the *necessary level of security*.

The **compliance with applicable transfer restrictions** at the recipient's location depends on legal regulations applicable to the transferred data types and may be restricted additionally by the constraints and agreements specified in the SLA of the service contract (cf. Section 3.2). To decide on applicable legal regulations and requirements of the SLA, security policies can be used (cf. Section 4.2.2). The enforcement takes place in the virtual resource scheduling of the cloud management process, as for example shown for the RESERVOIR architecture [171] by enforcing resource placement and transfer control for virtual machine migration under location constraints [139] [132]. The capabilities of virtual resource scheduling in the cloud management process and related approaches to consider location-determined data processing are investigated in more detail in Section 5.4.2.

To conclude, transfer control has to be implemented for data transfer and for the assignment and migration of virtual resources. The recipients' identity and authorisation have to be identified and verified, which can be achieved by using methods for identity and access management. Further, the *necessary level of security* has to be ensured at the recipients' location and applicable transfer restrictions have to be applied. This can be done by implementing decision and enforcement methods on the permissibility of data transfer within the cloud management process.

In this thesis, transfer control for the assignment of virtual resources is designed by incorporating the decision on the *ensured level of security* and the enforcement process for the *necessary level of security* and of the security polices into a single location-determined decision and enforcement process (cf. Section 5.4.2). Further, reliable methods for identifying the location of hardware resources and virtual resources are identified (cf. Section 5.4.1). To

describe the necessary information, an information model is developed using methods of information flow control (cf. Section 5.3). In the experimental evaluation, it is shown that transfer control can be implemented in cloud infrastructures exemplary for virtual machine assignment in OpenStack (cf. Section 6.1).

4.2.4 Counter measures and incident response

Cloud security management has to be able to deal with disturbances and irregular events (cf. Section 3.6.3.4). Such disturbances and irregular events can cover (1) unexpected or malicious behaviour of virtual resources, hardware resources and components of the cloud management process (e.g., compute server failure, maleware running on virtual machines, and software errors in the cloud management software), (2) unexpected or malicious behaviour of authorised entities (e.g., insider attack by employees of the cloud provider, unavailability of hosting sites, and denial of service (DoS) attacks performed by the cloud customer), and (3) unexpected or malicious behaviour of unauthorised third parties (e.g., intrusion of an external attacker). Cloud providers are obliged to implement mechanisms for preventing, detecting, and terminating infringements and malicious behaviour that they are aware of (cf. Section 3.6.3.4). To prevent and detect infringements and malicious behaviour, it is necessary to perform a risk analysis identifying threats to the security goal defined in the security policies, and specifically, possible vulnerabilities of used hardware and software. To terminate infringements and malicious behaviour, countermeasures and measures for incident response have to be defined by the cloud provider.

The literature identifies several threats, vulnerabilities, and potential countermeasures identified in cloud infrastructures [97] [89] [105] [212]. A major concern are threats to virtualisation security and vulnerabilities in virtual environments [97, pp. 5 et seqq.], which can be addressed in general by using trusted hypervisors [21] [179]. Other threats address data security, service availability, and identity and access violations [97, pp. 3 et seqq.], which can be addressed by implementing basic security measures (cf. Section 3.6.3.1) and identity and access management (cp Section 3.6.3.2).

In practice, there exist general recommendations on incident reporting [70]. Further, the IT security standards give specific recommendations and best practices on risk analysis, on countermeasures, and on incident response (e.g., German 'IT-Grundschutz' [31], ISO/IEC 27004 [111], and ISO 27005 [109]) However, there is an increasing number of publicly reported security incidents in existing cloud infrastructures [47], which indicates the need to further improve counter measures and incident response in practice.

In this thesis, the design and implementation of legally compliant data processing within cloud infrastructures is investigated with a focus on the challenge of location inhomogeneity (cf. Definition 4.6). Countermeasures and incident response are not part of the focus, and therefore are not further investigated. In Section 7.3, the support of countermeasures and incident response using the methods belonging to the approach presented in this thesis are discussed.

4.2.5 Conclusions on cloud security management

The analysis of existing approaches and practices on cloud security management reveals that the challenge of inhomogeneity (cf. Definition 4.6) is under-researched and lacks of comprehensive implementation in existing cloud infrastructures. The only existing approaches addressing the challenge are proposed by TClouds [180] and OPTIMIS [202], which are both recently completed European research projects on secure cloud computing. TClouds provides methods to securely identify the compute server hosting a virtual machine. OPTIMIS implements methods to automatically decide and enforce virtual resource assignments based on white-listing locations that satisfy the *necessary level of security*. None of the approaches provides an information model that is capable of dealing with requirements on applicable safeguards and security measures and particularly not with the *ensured level of security* at hosting sites. Such an information model is necessary for solving the challenge of inhomogeneity in general.

The analysis further reveals that there already exist several methods that can be applied to formulate security polices, implement security measures, and establish countermeasures and incident response mechanisms.

To address the challenge of location inhomogeneity in general, this thesis focuses on identifying and applying (existing and new) methods that are capable of solving the challenge. This includes the development of an information model particularly covering location information (cf. Section 5.3) and the extension of the cloud management process to support location-determined data processing (cf. Section 5.4.2). In particular, the incorporation of these methods into the cloud compliance management is investigated (cf. Sections 4.3 and 5.4.3).

4.3 Compliance management in the cloud

Compliance management has to ensure the effectiveness and implementation of safeguards that are necessary to satisfy legal requirements (cf. Section 3.6.4). Therefore, the cloud provider is required to (1) document the performed data processing and applied safeguards, (2) monitor the effectiveness of safeguards and satisfaction of legal requirements, and (3) report on legal compliance to the cloud customer.

In particular, compliance management of the cloud provider has to ensure the effective security and compliance at the hosting site by communicating applicable security policies and monitoring legal compliance at hosting sites.

In this section the technical capacity to implement compliance management in cloud computing is investigated. In Section 4.3.1, methods for logging and documentation of safeguards and data processing in cloud infrastructures are analysed. Approaches on compliance monitoring for cloud computing (cf. Section 4.3.2) and the capabilities and trustworthiness of compliance reporting (cf. Section 4.3.3) are then explored.

4.3.1 Logging and documentation

The documentation of preformed data processing and applied safeguards is necessary to provide an information basis for compliance monitoring and reporting (cf. Section 3.6.4). Relevant

information for documentation covers in particular the events and location of data processing and storing, attempted and granted access (to cloud management, virtual resources and hardware resources), and administrator activities (of cloud customer and cloud provider).

In cloud infrastructures, documentation can be implemented on the level of (i) cloud management, (ii) virtual resources, and (iii) hardware resources. Table 4.2 provides an overview of possible information sources and the items that can be documented at each level, which are discussed in the following.

On the cloud management level, it is possible to document the management actions of cloud customers and cloud provider as well as the behaviour, configuration, and status of the cloud infrastructure. In particular, documentation of communication with the management interface provides evidence of management actions performed by the cloud customer and cloud provider and of their information basis (which is provided by the management front-end and used for management decisions). The same information can be documented for communication with third party cloud infrastructures. Such information is important for clarifying the responsibility for the configuration of virtual resources and their applied security polices. Further, it is possible to gather information on virtual resources that are visible to the cloud management (e.g., virtual resource identifier, resource configuration, and assigned hosting site). Such information can be used to verify the compliance of management functions and can be compared with information gathered on the virtual resource layer and the hardware resource layer for plausibility checks. However, the value of evidence of information provided by management interfaces is limited, since the source of information is not verified. For example, the information can be taken from the cloud management databases, like the nova database on resources and their state.[1]

On the virtual resource level, the interaction of end systems located outside the cloud with virtual resources located inside the cloud can be documented. This is needed to supervise access to virtual resources and data transfer to end systems. Additionally, logging mechanisms of guest OS and applications running on virtual resources can be used to document the data processing in virtual resources and applied security measures. In IaaS, these logging mechanisms are operated on behalf and within the area of responsibility of the cloud customers and are not available to the cloud provider. The same holds true for communication between virtual resources via the cloud customer's VPN. The cloud customer can decide to provide the logging and communication information to the cloud provider, and additionally, can use it for comparison with information provided by the cloud provider for plausibility checks.

On the hardware resource level, it is possible to document the operation of the hardware resources and the hypervisor. The hardware resources and hypervisor are both operated at the hosting site. Therefore, the visibility of their operation at the cloud management level depends on the information provided by the hosting sites (for example, via the VMI), and consequently, can vary in granularity and trustworthiness for different hosting sites. The operation of hardware resources and the hypervisor are not visible at the virtual resource level, because they are obscured by the resource virtualisation. Therefore, both the hardware resources and the hypervisor are important information sources for documenting data processing in the cloud. In

[1] OpenStack wiki entry on 'HAforNovaDB' (high availability for Nova database), on the Internet: https://wiki.openstack.org/wiki/HAforNovaDB (last visited: 30.06.2015).

Table 4.2: Documentation in cloud infrastructures

Level	Information sources	Items of documentation
Cloud management	Communication with management front-ends, hosting sites, and third party cloud infrastructures	• Triggered actions • Requested virtual resource configuration and security policy • Reported feedback • Applied communication security • Actor's identifier used for authentication
Cloud management	Information provided by the management interfaces (i.e., CMI, FMI, and VMI)	• Configuration and status of virtual resources • Configuration and status of cloud management components • Communication endpoints • Identifier of virtual resources
Cloud management	Control messages of the management interfaces	• Actions triggered and performed by using the management interface • Response messages of the connected systems • Identifier of virtual resources
Virtual resources	Communication with end systems	• Access (attempts and granted) to virtual resources • Data transfer to end systems • Applied communication security • Communication endpoints • Actor's identifier used for authentication
Virtual resources	Logs generated within virtual resources (e.g., by guest OS and applications running on virtual resources)	• Actions triggered and performed by guest OS and applications (e.g., data processing) • Configuration and status of guest OS, applications, and security measures within the virtual resources
Virtual resources	Communication between virtual resources	• Actions triggered and performed by virtual resources (and hosted guest OS and applications) • Response messages of virtual resources • Data transfer • Applied communication security • Communication endpoints • Identifier of involved virtual resources
Hardware resources	Communication with end systems	• Applied communication security • Identifier of involved end systems • Communication endpoints • Actor's identifier used for authentication
Hardware resources	Information provided by the hardware management interfaces	• Configuration and status of hardware resources
Hardware resources	Information provided by the hypervisor	• Configuration and status of virtual resources • Identifier of hardware and virtual resources
Hardware resources	Logs generated on hardware resources (e.g., by OS and applications running on hardware resources)	• Actions triggered and performed by OS and applications (e.g., data processing) • Configuration and status of OS, applications, and security measures running on the hardware resources
Hardware resources	communication between hardware resources	• Actions triggered and performed by hardware (and hosted OS and applications) • Response messages of hardware resources • Applied communication security • Communication endpoints

particular, the information on their operation can be used to identify the allocated hardware resources that are used for operating virtual resources. Information on resource allocation allows the verification of the location of data processing and the *effective level of security*, which are both important for legally compliant data processing (cf. Section 4.2.1).

Other sources of information are vulnerability analyses performed on hardware resource and virtual resource level as well as the manual entry of information by involved parties (e.g., cloud customer and cloud provider) [207]. Neither is limited to a specific level of cloud infrastructures.

A vulnerability analysis allows the inspection of the effectiveness of implemented security measures, and therefore, provides an information basis on which to verify and extend documentation on security measures created by the logging mechanisms mentioned above. There exist frameworks for systematic and comparable vulnerability analyses [101] [216] [155], and it is possible to perform vulnerability analyses on an automated basis (e.g., by using OpenVAS [168]). By using existing frameworks and tools, vulnerability analyses can be considered a reliable and comprehensive method for gathering information on the effectiveness of implemented security measures.

Manual entry of compliance relevant information provides the opportunity to document additional information that is not directly assessable from the cloud infrastructure (e.g., purpose of data processing and contract information). Further, the information can be used for plausibility checks of assessable information in the cloud infrastructure (e.g., the geo-location of hosting sites and customers' requirements for data processing). The reliability of manual entries is limited to the correctness of the entered information. Due to media discontinuity, there can be transcription errors and misrepresentation (fraudulently or accidentally). Double checking manual entries, in particular by multiple parties (e.g., cloud provider and cloud customer), can help to detect transcription errors and misrepresentation. However, transcription errors and misrepresentation can remain undetected, and therefore, are item of the trust relations between the involved parties (particularly between the cloud provider and cloud customer).

4.3.2 Compliance monitoring

The purpose of compliance monitoring is to observe and validate the effectiveness and implementation of safeguards with respect to legal and contractual requirements (cf. Section 3.6.4). For that purpose, documentation (cf. Section 4.3.1) and applicable security policies (cf. Section 3.6.2) are used as an information basis.

To monitor legal compliance, cloud providers have to consider the cloud infrastructure operated by themselves and the hardware resources of subcontracted data hosting sites and third party cloud infrastructures (cf. Section 3.6.4). Therefore, compliance monitoring of the cloud provider relies on documentation and compliance reports provided by hosting sites and third party cloud infrastructures.

The literature, gives different approaches on compliance monitoring that can be classified into (i) *manual validation* performed by a human person, (ii) *semi-automated validation* performed by a human person using supporting tools, and (iii) *fully automated validation* without human interaction.

For **manual validation**, standard procedures are described in the information security management standards using life-circle approaches, for example, described by the ISO/IEC 27001 standard [112] and by the German 'IT-Grundschutz' [31]. While procedures on manual validation are well understood and standardised, they are limited by the human capabilities to observe and validate information. Therefore, manual validation does not scale in the same way as cloud infrastructure and service provision do which are both designed for automated and highly scalable data processing in general. In particular, manual validation is not feasable for validation at run-time since automated interaction in clouds is beyond human recognition. However, manual validation is suitable for initial and periodic validation of overall security and compliance in cloud infrastructures.

To address scalability, there exist **semi-automated approaches** on compliance monitoring that can gather and pre-process the data needed to monitor compliance and to present it in a legible form for further validation. Examples of such approaches are known in the context of privacy protection in Web Services [225] and traditional IT outsourcing [93]. Existing approaches are also applicable to cloud computing, for example, to perform a privacy impact assessment [195] or to verify the trustworthiness of provided logging data [144]. An other approach to support scalability is to perform the validation on randomly selected examples [144]. This reduces the effort for validation to the number of selected examples, and therefore scales by reduction of the density of selected examples. An advantage of this method is the support of the principles of data reduction and data economy in data protection law (e.g., German data protection law, BDSG §3). A drawback of this method is that there remain invalidated 'blind spots' and that reducing the density of selected examples results in the reduction of the evidence provided, since the number of 'blind spots' increases. This drawback can be mitigated by storing the documentation of 'blind spots' for later validation, for example, to provide evidence on specific data processing to the cloud customer. In general, semi-automated approaches scale with the data processing, but they require further validation by a human being, and therefore are not applicable for run-time validation.

To address validation at run-time, there exists **fully automated approaches** on compliance monitoring considering run-time information, for example, by introducing automated validation of SLA parameters by evaluating the operational state of virtual resources and hardware resources at run-time [62] [63]. Fully automated compliance monitoring is capable of validating at run-time and additionally scales with the data processing. However, it depends greatly on the correctness and granularity of the requirements defined within the security policies and the information documented. For example, the possibility of implausible requirements and documentation has to be considered when designing the monitoring mechanisms, because additional checks and validation are required to detect them. While it is possible to detect implausibilities in documentation by using different sources of information (e.g., log-files of hypervisors and control data of VMIs), the translation of legal requirements to security policies is often error-prone [158]. Here, the manual validation by a human being has the advantage that inconsistencies and implausibilities can be detected more flexibly and during the evaluation process. Another approach to ensure the correctness of the documentation is using remote attestation in trusted infrastructure [180, pp. 87 et seqq.]. By using a physical trust anchor (e.g., TPM [203]), it is possible to implement trusted hypervisors that can remotely attest to the integrity of the

program code executed and to the integrity of the particular virtual machines being operated [79]. Assuming the correctness of the program code deployed, the remote attestation provides strong evidence of the correctness of the information provided by the hypervisor (including the documentation of virtual machine operation).

In practice, compliance monitoring is addressed in IT security standards, for example, ISO/IEC 27001 [112] and the German 'IT-Grundschutz' [31]. Additionally, there exist recommendations on compliance monitoring provided by security authorities, for example, a checklist provided by ENISA [68] and guidelines on continuous monitoring provided by NIST [51]. There also exist approaches showing that it is possible to implement compliance monitoring in existing cloud infrastructures, for example, semi-automated monitoring of SLA compliance [202, pp. 22 et seqq.], semi-automated validation of compute node integrity in OpenStack [53, pp. 12 et seqq.] [53, p. 60], and fully automated validation of confidentiality and integrity using simulated remote attestation [116].

To conclude, there exist several methods, best practices, and implementations for compliance monitoring in cloud infrastructures, but none of these addresses the challenge of location inhomogeneity by monitoring the *effective level of security* (cf. Section 4.2.1). Nonetheless, remote attestation of trusted hypervisors can be used for semi- and fully automated validation and provides strong evidence of the monitored operations.

In this thesis, monitoring the *effective level of security* is addressed by proposing a semi-automated approach to monitoring the compliance of virtual resource placement using hypervisor log validation (cf. Section 5.4.3). Additionally, the capabilities and the limits of compliance monitoring are reflected in the context of legal evaluation in Section 6.2.1 and technical evaluation in Section 6.2.3.

4.3.3 Compliance reporting

In the context of IT outsourcing to the cloud, the purpose of compliance reporting by the cloud provider is to provide information and evidence on the compliance of data processing that is performed on behalf of the cloud customer (cf. 3.6.4). These reports are required by the cloud customer to verify the compliance with legal requirements and SLAs assured by the cloud provider. The information basis for compliance reports consists of documentation on the processing of the cloud customer's data and the cloud customer's security policies (i.e., SLAs and contractual agreements). In particular, information on data processing and its compliance (with the cloud customer's requirements) at subcontracted hosting sites and third party cloud providers have to be considered. In general, the reliability and correctness of compliance reporting are important, since the cloud customers rely on these reports to evaluate the legal compliance of data processing in the cloud.

The reporting on the performed data processing and on its compliance with applicable security requirements can be performed – similarly to the validation in compliance monitoring – as follows: (i) manually by a human beeing, (ii) in a semi-automated ways by a person using supporting tools, and (iii) in a fully automated way without human interaction. Each of these methods has the same implications for scalability and applicability at run-time as the respective methods have for compliance monitoring (cf. Section 4.3.2). In the following, examples of compliance reporting in the literature and in practice are discussed.

In the literature, manual compliance reporting is addressed, for example, when the industrial standards on service operation controls SSAE 16 [7] and the ISAE 3402 [107] define an audit process performed by a trusted third party (i.e., certified auditor). The scope of the standards is financial and IT-security-related reporting in outsourcing scenarios and is applicable to cloud computing, too. In particular, the Service Organization Controls (SOC) 2 report of the SSAE 16 standard considers the reporting on cloud computing [8, pp. 141 et seqq.].

Approaches to semi- and fully automated compliance reporting make use of reporting services that provide compliance information that can be evaluated by a human being. For example, it is possible to give fully automated reports on SLA violations [25] [63] and on privacy protection and transfer control in orchestrated web services [99].

In practice, existing cloud infrastructures usually do not offer compliance reporting. Rather, the opposite is the case. For example, in the Amazon EC2 Service Level Agreements,[1] breaches of the assured availability have to be reported by the cloud customers in order to get refunded.

An example of implemented compliance reporting is the reporting on data location compliance in the public sector using the RESERVOIR architecture [139]. In this approach, monitoring information on virtual storage is enriched with information on the assigned hosting site. Then, this information is associated with the geographical location of the hosting site. The approach assumes that the hosting sites and cloud provider are cooperative and does not provide any mechanisms to ensure the correctness and reliability of the reported information. Using this approach, it is possible to verify whether data are processed within a specific country or not.

To conclude, there exist methods for compliance reporting applicable to cloud infrastructures, in particular for reporting on SLA compliance and violations. Further, it is possible to offer fully automated reports on data location compliance. In any case, compliance reporting is not usually implemented in existing cloud infrastructures and existing approaches do not address the reporting on legal compliance, and particularly not on compliance with the *necessary level of security*.

In this thesis, compliance reporting on the *necessary level of security* is addressed by proposing a fully automated approach to visualising the virtual machine placement within the cloud using a geographical map.

4.3.4 Conclusions on compliance management

The analysis of compliance management in cloud infrastructures reveals that the primary challenges are the acquisition and validation of telemetry data. Both have to scale with the number of virtual resources requested by the corporate customer and with the amount of data processing on behalf of the corporate customer. Therefore, manual monitoring and reporting methods are not suitable and at least semi-automated approaches are required. Existing guidelines on compliance monitoring and reporting in security standards focus on manual validation (e.g., SOC 2 reporting). Using semi-automated approaches allows the enrichment of the manual validation process with automated tool support on information acquisition and validation. When

[1] Amazon EC2 Service Level Agreement (Effective Date: June 1, 2013), on the Internet: `http://aws.amazon.com/ec2/sla/` (last visited: 30.06.2015).

doing so, the reliability and correctness of information provided by these automated tools is paramount.

Further, an important source of information are hypervisors, since they can provide reliable information on virtual resource operation. Correctness of the information can be ensured by using, for example, the remote attestation of trusted hypervisors.

In general, there exist approaches on compliance monitoring and reporting in cloud infrastructures, but compliance monitoring with respect to the *necessary level of security* by the cloud provider and the reporting of this compliance to the corporate customer remains an unsolved challenge. In this thesis, the gap is addressed by providing a documentation model and identifying monitoring and reporting methods for compliance control (cf. Section 5.4.3). Further, the applicability of the proposed model and used methods is evaluated in an experimental evaluation using a proof-of-concept implementation (cf. Section 6.1). In additional, the capabilities and limits of compliance monitoring and reporting are discussed in Section 6.2.3).

4.4 Conclusions on implementing legally compliant clouds

In this chapter, the technical capability of clouds was analysed with respect to the implementation of the legal requirements identified in the legal analysis (cf. Section 3.6). In a first step, an entity-relationship model of IaaS clouds was formulated using the formal notation of an ontology. This model provides a formal basis for describing clouds. It was used in the analysis made in this chapter and will be used in Chapter 5 to identify and describe information flows in clouds. In a second and third step, the technical capabilities of cloud security management and compliance management in clouds were analysed and the research gap in addressing legal requirements was identified. In this context, the challenge of location inhomogeneity was defined (cf. Def. 4.6. It describes the key challenge of this thesis: to address the legal necessity of location-determined data processing in clouds (cf. Section 3.5.3) technically. The analysis of cloud security management reveals that current research is aware of this challenge, but solutions proposed thus far do not address the challenge or only part of it (cf. Section 4.2.1). In particular, the need for an information model that is able to describe legal constraints on location and security in a technically decidable and enforceable way remains unaddressed. Other aspects of legal compliance, like security policies (cf. 4.2.2) and security measures (cf. 4.2.3) were thoroughly discussed, but the challenge of location inhomogeneity remains to be addressed. The analysis of compliance management in clouds revealed that current cloud telemetry is able to monitor data processing in the cloud (cf. Section 4.3.1), but lacks automated compliance monitoring (cf. Section 4.3.2) and compliance reporting (Section 4.3.3). Both the missing information model and automated compliance monitoring and reporting are addressed in Chapter 5, where a model for information flow control is proposed, tackling the challenge of location inhomogeneity in clouds.

Chapter 5

Tackling location inhomogeneity with information flow control

In this chapter the challenge of location inhomogeneity (cf. Definition 4.6) is addressed by introducing information flow control to the cloud management process. In this connection, data transmissions are considered information flows, and transmission control equates to information flow control. To address the security goals identified in the legal analysis (cf. necessary safeguards at the cloud provider in Section 3.3), a model of information flow control in clouds is described addressing confidentiality, integrity, availability, and location determination (Objective 5). Further, the implementation of information flow control within the cloud resource management is proposed in a way so that it (i) introduces location-determined data processing with respect to cloud resources and (ii) enables the monitoring and reporting of compliance with respect to the modelled security goals (Objective 6).

To achieve this, we need to understand the information flow in cloud infrastructures first. Based on this knowledge, the information flow that is relevant for tackling the location inhomogeneity can be identified and modelled. This requires the transition of existing methods to clouds since information flow control in this context is a novel approach. Particularly, modelling availability and location have not been addressed in this connection. Therefore, existing methods have to be extended by these security properties. For the practical application of information flow control, it is further necessary to establish a reliable instance that is able to enforce information flow control within the system. Here, the cloud management process (as described in Section 4.1.4.4) is identified as a good candidate for doing so since it assigns virtual resources to hardware resources and therefore has control of data transmissions triggered by virtual resource placement and migration. It is shown that information flow control can be implemented in the cloud management process and that it is possible to establish a trustworthy resource classification as well as effective compliance monitoring and reporting in clouds.

In Section 5.1, information flow in cloud infrastructures is analysed and requirements for modelling information flow in clouds are identified. In particular, the major importance of controlling information flow caused by virtual resource placement and hardware resource allocation is identified, which is the focus of this thesis.

In Section 5.2, the necessary background on information flow control is provided. After

motivating the decision to select the methods of mandatory access control, existing models for lattice-based access control are described and their issues when it comes to tackling location inhomogeneity in clouds are discussed.

In Section 5.3, a generalised model for access control is introduced that incorporates the existing models described in Section 5.2 and, in addition, is able to model information flow based on location and availability constraints. Further, the application of this model to the cloud management process is described.

In Section 5.4, the implementation of information flow control in clouds is investigated with the focus on location-determined data processing. In particular, possible approaches to trustworthy handling of location information, to location-determined resource management, and to compliance monitoring and reporting are discussed.

In Section 5.5, the lessons learned with relation to tackling location inhomogeneity in clouds are concluded.

5.1 Information flow in cloud infrastructures

Before modelling information flow control in cloud infrastructures, information flow in clouds has to be understood. Beside the information flow in and between the corporate customer's applications running on virtual resources, there is information flow between virtual and hardware resources implicated by virtual resource placement of the cloud management process. For example, whenever a virtual resource is assigned to a specific hardware resource, the information contained within the virtual resource flows to the assigned hardware resource. Such information flow has to be considered particularly when implementing information flow control in cloud infrastructures, since it is usually not visible to corporate customers due to the hardware resource abstraction of virtualisation.

In a case of IT outsourcing to the cloud, the contracted relationship between corporate customer and the cloud provider specifies whether information flow is allowed or not. For example, the information flow between the virtual resources of a single the corporate customer is usually allowed, since they are operated on the behalf of the corporate customer and therefore information remains in the corporate customer's authority. Then again, the information flow from a German hosting site to a French hosting site is forbidden if the corporate customer is allowed to process data only within Germany.

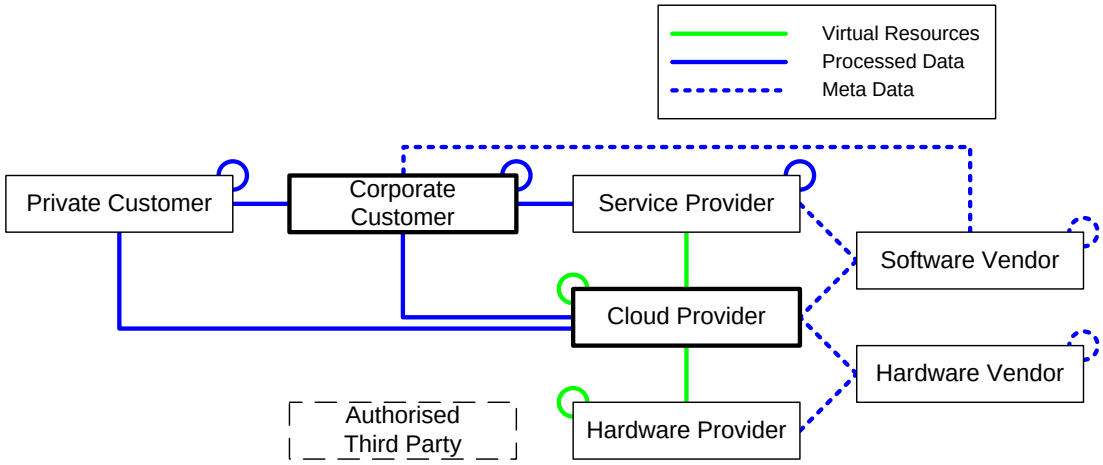
Further, it can be observed that there is a separation of responsibility for technically controlling the information flow. On the one hand, the corporate customers control the information flow from and to their virtual resources, and the cloud provider has little to no influence on the information flow on this level. On the other hand, the cloud provider controls the resource management in the cloud, i.e., virtual resource placement and hardware allocation, which generally triggers information flow in the cloud. As a result, the corporate customer has little to no influence over the information flow on the level of cloud resource management. For instance, the cloud provider is responsible for moving a virtual resource from a German hosting site to a French hosting site, while the corporate customer is responsible for running an application moving data from a virtual resource hosted in Germany to a virtual resource hosted in France. This indicates that it is necessary to model information flow control on two different levels: (1)

on the level of data processing by corporate customers and (2) on the level of cloud resource management at the cloud provider. Further, it is possible that other actors can cause information flow in clouds, which has to be clarified. For example, if a service provider operates an application that interacts with a database then there is information flow between the application and the database. The application and the database do not necessarily have to be running on the same virtual resource and, therefore, may be hosted on different hardware resources.

In Section 5.1.1, the information flow in cloud infrastructures and particularly for IT outsourcing to cloud infrastructures is investigated. Based on the involved entities (i.e., actors) described in Section 2.2, the existing types of information flow are identified, and allowed and forbidden information flow is classified. Then, in Section 5.1.2, the areas of responsibility of corporate customers and cloud providers are identified. In particular, the impact on information flow control is investigated, which identifies prerequisites for modelling the separation of responsibilities while technically enforcing information flow control. In conclusion, the assumptions that are made for modelling information flow control in clouds in this thesis are provided in Section 5.1.3.

5.1.1 Information flow in IaaS cloud computing

Figure 5.1: Information flow between legitimate actors (differentiated by virtual resources, processed data, and meta data). The depiction of information flow with authorised third parties is omitted since they can interact with any other actor.



There are three different types of information flow that can be identified between the legitimate actors. Figure 5.1 illustrates the three types of information and the way they flow between the different actors. The different types of information cover (1) the data processed within the cloud, (2) the virtual resources operated in the cloud, and (3) meta data of hardware and software, which are exchanged during their operation. For authorised third parties the information flow depends on which basis of authorisation the third party acts on and with whom the third party interacts. For example in a case of full inspection, there can be all three types of infor-

mation flow. For that reason, the depiction of the information flow for authorised third parties is omitted in Figure 5.1. Further, it is assumed in this thesis that for authorised third parties all three types of information flow are allowed. For a more differentiated control of information flow to authorised third parties, it is necessary to extend the model of information flow by conditional constraints indicated under which circumstances a specific information flow is allowed. This requires additional research with a focus on information flow to authorised third parties, which is not part of this thesis. Possible extensions are discussed in Section 7.3. In the following, each type of information flow is discussed.

The first type of information flow (processed data) is related to the customers' data that are processed within the cloud. The data are provided by the private and corporate customers and are processed on their behalf by the service and cloud providers. Further, the data can be exchanged with others customers, e.g., in a Business-to-Business (B2B) scenario between two corporate customers, and also between service providers when providing subcontracted cloud services, e.g., when using a virus scanning service for an email service. In IaaS, the customer has direct influence on the information flow of the processed data, since the data processing application are under the customer's control. The providers can also have an influence, since they are operating the resources running the applications. However, except for the service provider, the providers do not directly interact with the data processing applications.

The second type of information flow (virtual resources) is related to the virtual resources that are operated in the cloud. The information flow of virtual resources differs from that of processed data, since virtual resources represent a part of the data processing system itself (cf. Section 4.1.2). Each virtual resource can contain customers' data (i.e., processed data) mentioned in the first type of information flow. Without inspecting the virtual resources, the data processed within remain hidden inside the virtual resources and beyond the awareness of the cloud, hardware, and service providers. However, if the virtual resource is migrated, the data within are migrated too, resulting in an information flow of processed data. There same is true for initial virtual resource placement, if the customers' data are contained in a virtual resource (e.g., in the case of customer defined virtual machine images). Also, the destruction of a virtual resource usually results in the deletion of contained data.[1] Therefore, the information flow of virtual resources usually also covers the information flow of processed data. To conclude, virtual resources are operated under the control of a service provider, a cloud provider, and a hardware provider and are exchanged among them (e.g., due to resource migration). The customers have no direct influence on this type of information flow.

The third type of information flow (meta data) is related to different types of meta data that are exchanged with the software and hardware vendors. Meta data in this context are data that is directly related to the software and hardware itself and usually covers information on software execution and hardware operation. Examples for meta data are license information, usage statistics, and error reports on failure. Usually, meta data do not contain customers' data. However, it is possible for meta data to contain customers' data. For example, a memory snapshot after a hardware failure may contain customers' data that was loaded into the memory. What

[1] In this thesis it is assumed that on destruction the allocated hardware resources are freed and no longer can be addressed by the corresponding customers. However, data are not necessarily deleted or overwritten immediately after freeing the hardware resource and might be recovered on physical access.

types of data are transferred to the vendor can vary for different vendors and applications, and they have to be investigated for every single case. Often, the customers and providers can directly influence the transmission of meta data, for example, in the case of optional transmission of error reports or by filtering the transmitted data. If such options are consequently implemented, the information flow of processed data as a part of the meta data can be fully controlled by the customers and providers, and therefore be handled like the information flow of processed data. In this thesis it is assumed that the information flow of meta data does not contain processed data, and it will therefore not be further investigated. Possible extensions of the information model for covering meta data are discussed in Section 7.3.

To summarize, the information flows of processed data and of virtual resources are most relevant for compliant data processing in the cloud, since both types of information flow contain customers' data that have to be processed in a legally compliant manner and according to the customers' mandate. In this thesis following definitions of information flow of processed data and of virtual resources are made according the observations above.

Definition 5.1 (Information flow of processed data) *All types of data that are processed on behalf of a corporate customer within a virtual resource (of the cloud) a considered to be* processed data*. Further, processed data are associated with the corporate customer on whose behalf they are processed. Then, the* information flow of processed data *is considered the access of an actor (cf. Section 2.2.1), virtual resource (cf. Section 4.1.2), or hardware resource (cf. Section 4.1.3) to the processed data.*

Definition 5.2 (Information flow of virtual resources) *The term* virtual resources *is used according to the entity-relationship model on IaaS clouds defined in Section 4.1 and covers virtual machines, virtual storage, virtual links and virtual network services (cf. Section 4.1.2). The* information flow of virtual resources *is considered the access of an actor (cf. Section 2.2.1) or hardware resource (cf. Section 4.1.3) to the virtual resources, for example, due to virtual resource placement or migration to a specific hardware resource.*

Remark 5.1 (Accessing of vs. connecting with virtual resources) *There is a difference between accessing a virtual resource (e.g., a server executes a virtual machine) and connecting with a virtual resource (e.g., a corporate customer has a network connection with a virtual machine). While the first is considered the information flow of virtual resources, the latter is considered the information flow of processed data. In this thesis, accessing and connecting are distinguished form one antother in that* accessing of virtual resources *means that virtual resources are fully accessible on the virtualisation level and* connecting with virtual resources *means that a network connection is established with a network endpoint of the virtual machine.*

Remark 5.2 (Equivalence of access on virtual resources) *The legitimate actors access virtual resources via a hardware resource that hosts the virtual resources. For example, hardware providers access virtual machines when they are migrated to one of their servers. Another example is the duplication of virtual storage for a backup located at a third party cloud provider. Hardware resources access virtual resources when hosting them. In general, for every hardware resource there is a legitimate actor that is responsible for its operation. Since legitimate actors access virtual resources via a hardware resource, the access of a legitimate actor can*

always be described by an equivalent access of the involved hardware resource and vice versa. This equivalence makes it possible to model the information flow of virtual resources on the basis of hardware resources only.

Further, it is observed that the different types of information flow cannot be controlled by every actor equally. While the information flow of processed data is primarily controlled by the customers, the information flow of virtual resources is primarily controlled by the cloud and the hardware providers. This has a significant impact on how information flow control has to be implemented in the scenario of IT outsourcing to the cloud. This is because the control mechanisms have to be operated by different parties. As a result, the actions of these parties have to be coordinated (to some extent) to achieve the overall security goal: dealing with the challenge of location inhomogeneity (cf. Def. 4.6). In Section 5.1.2, the impact of this separation of responsibility and control is investigated, and how it has to be addressed in the information model.

5.1.2 Separation of responsibility and information flow control

In Section 5.1.1, *information flow of processed data* (cf. Def. 5.1) and *information flow of virtual resources* (cf. Def. 5.2) were identified. To address the challenge of location inhomogeneity (cf. Def. 4.6), it is important that both types of information flow can be controlled with respect to the origin and the target location. Further, information flow is allowed only between legitimate actors that are authorised to access processed data or virtual resources. In particular, the actors have to ensure the *necessary level of security* (cf. 3.6.1). Therefore, it is necessary to control the *information flow of processed data* and the *information flow of virtual resources* on the basis of target location, involved actors, as well as the *necessary level of security* and the *ensured level of security*. As noted in Section 2.2, the *information flow of processed data* is primarily under the control of the corporate customers, and the *information flow of virtual resources* is primarily under the control of the cloud and hardware providers. Additionally, cloud providers and corporate customers decide on their own which actors are involved in the data processing, and often the cloud providers have little or no knowledge of the actors involved by the corporate customers and vice versa. The question arises as to what impact this separation of responsibility and control has on the information model.

Due to the separation of responsibility and control, the *information flow of processed data* and the *information flow of virtual resources* can be expressed at two different levels of information flow, and it is possible to investigate whether there is information flow between these two levels. First of all, virtual resources may contain processed data, since the processing of data is the purpose of virtual resources. This implies that if there is *information flow of virtual resources* (e.g., virtual resources are migrated) then there is also *information flow of processed data*. Conversely, the *information flow of processed data* does not result in that of virtual resources. However, the *information flow of processed data* can change the state of a virtual resource with respect to whether specific data is contained in a virtual resource or not. This change of state is important if the hosting location of a virtual resource is decided by the data processed within a virtual resource. On the other hand, the *information flow of virtual resources* does not change the data processed within the virtual resources. However, in the context of in-

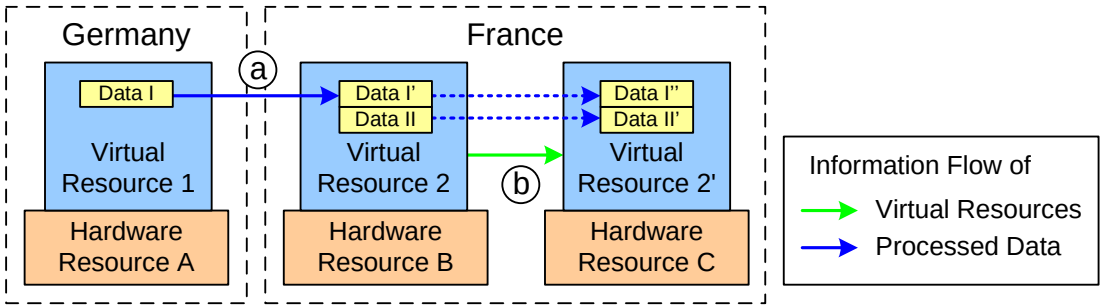
formation flow, it can change the state of the processed data with respect to location, involved actors and *ensured level of security*.

The results are summarised in in three observations:

- **(Observation 1)** The *information flow of virtual resources* affects that of processed data, and particularly, it can change the state of processed data with respect to location, involved actors, and the *ensured level of security*.
- **(Observation 2)** The *information flow of processed data* does not affect that of virtual resources.
- **(Observation 3)** The *information flow of processed data* can change the state of virtual resources with respect to data processed within the virtual resources.

Figure 5.2 exemplifies the observations made on information flow in a scenario with three hardware resources that are located in two different countries. There are two successive actions (a) and (b) which result in information flow. In the first action (a) *Data I* is copied from *Virtual Resource 1* to *Virtual Resource 2*, which is an *information flow of processed data*. Copying data does not result in an *information flow of virtual resources* (Observation 2). However, *Virtual Resource 2* now contains *Data I'* in addition to *Data II*, which is a change of its state with respect to data processed within *Virtual Resource 2* (Observation 3). In the second action (b) *Virtual Resource 2* is copied from *Hardware Resource B* to *Hardware Resource C*, which is an *information flow of virtual resources*. By copying the *Virtual Resource 2*, *Data I'* and *Data II* are copied and now located on *Hardware Resource C*, which is an *information flow of processed data* that is triggered by an information flow of virtual resources (Observation 1). From the perspective of information flow control, there are two questions arising from this example. First, if corporate customers trigger action (a), how do they know that their data are copied cross-boarder to another country? Second, if cloud providers trigger action (b), how do they know whether the data processed in virtual resources has been allowed to be copied to the target hardware resources or not? Both questions are explored in a more general context subsequently.

Figure 5.2: Example of interferences in the information flow between three hardware resources located in Germany and France



Applying the three observations to the IT outsourcing scenario provides better insight into the information flow controlled by corporate customers and by cloud providers. In the IT outsourcing scenario, the corporate customers are responsible for outsourcing data processing to the cloud. Therefore, it is important for them to take control of the *information flow of processed data*. By configuring and controlling the operating systems and applications running on the virtual resources, the corporate customers can take control of the *information flow of processed data* that is initiated by the operating systems and applications. However, according to Observation 1, the *information flow of processed data* is influenced by that of virtual resources, and the latter is under the control of the cloud provider. This implies that the corporate customers have to rely on the cloud provider and the cloud provider controls the *information flow of virtual resources* according to their needs (i.e., in compliances with legal and contractual obligations on data processing). Otherwise, the customers' efforts to control the *information flow of processed data* can fail through side-channels created by the *information flow of virtual resources*. Inversely, due to Observation 2, the *information flow of virtual resources* is not influenced by that of processed data. Thus, the cloud provider does not necessarily need to consider the *information flow of processed data* when controlling that of virtual resources. This makes the *information flow of virtual resources* a good candidate for establishing the legally required transmission control in clouds (cf. Section 3.6.3.3), since it cannot be overwritten by the *information flow of processed data* but can result in the latter.

To support the corporate customers' control over the *information flow of processed data*, the cloud provider needs to know what *information flow of virtual resources* complies with the transmission control requested by the corporate customer. Therefore, an initial agreement about allowed information flow is necessary. Such an agreement can be made on the basis of classifying the data contained within a virtual resource. Based on that, the cloud provider can decide whether *information flow of virtual resources* is allowed or not. However, due to Observation 3, it is necessary to consider continuously what types of data are contained within a specific virtual resource, since this can change due to the *information flow of processed data*. This can be addressed by classifying virtual resources based on the types of data they process. For example, a virtual resource is classified as be usable only for processing personal data originating from Germany. Further, this virtual resource is to be hosted only by hardware providers that are allowed to process personal data originating from Germany. Let's assume that there are multiple virtual resources that are classified as being usable only for processing personal data originating from Germany. Then, the corporate customer can allow information flow between these virtual resources and the information flow complies with the requirements for processing personal data originating from Germany.

5.1.3 Conclusions on modelling information flow

In this thesis, a model for information flow control in clouds based on the classification of virtual resources is developed. In this section, the modelling assumptions and prerequisites are summarised.

Assumption 1: *Multiple corporate customers operate concurrently on a cloud infrastructures with multi-level classification.*
Evidence: According to the IT outsourcing scenario, it is reasonable to assume that a cloud

provider is contracted by multiple corporate customers to provision cloud resources (cf. Section 2.2.1). Consequently, there are multiple corporate customers operating concurrently on cloud infrastructures. Further, for practical and security reasons, a cloud infrastructure has to be able to distinguish the requests and virtual resources of the different corporate customers. This property is called *multitenancy*. Additionally, each corporate customers can have different security requirements, and therefore the *necessary level of security* can be different, too (cf. Section 3.6.1). This implies that a cloud infrastructure has to be able to handle these different levels of security requirement and, thus, has multi-level classification with respect to the security requirements.

Assumption 2: *A cloud infrastructure is operated by a single cloud provider and hosted at multiple hosting sites, which are operated by hardware providers (one hardware provider per hosting site).*
Evidence: In this thesis, global clouds are investigated, which are operated in multiple countries (cf. Def. 2.5). This implies that there exist multiple hosting sites that are located in different countries. Explicitly, national clouds and particularly clouds with only a single hosting site are excluded by this assumption. They are not in the scope of this thesis, since they are generally location homogeneous (cf. Def. 2.2 and usually there is no need for transmission control within the cloud. Further, for practical reasons, the operation of a cloud infrastructure requires a single responsible entity coordinating the operation of hardware and virtual resources. Additionally, such an entity is the contractual partner of corporate customers and is liable for executing the contracted IT outsourcing in a legally compliantly manner. However, in a global cloud, there might be multiple national cloud providers operating each a local/global cluster of the cloud (cf. Section 4.1.4). In any case, there is only a single cloud provider that is operating the *Cloud Service Fabric* (cf. Section 4.1.4), since there is only one per cloud infrastructure. Otherwise, the existence of multiple *Cloud Service Fabrics* implies that there are multiple cloud infrastructures. Therefore, the cloud provider operating the *Cloud Service Fabric* is considered the cloud provider operating the cloud infrastructure. For analogous reasons, it is assumed that there is only a single responsible entity (i.e., hardware provider) for each hosting site, which is the contractual partner of the cloud provider. Having a single responsible entity for clouds and hosting sites is more structuring than restricting.

Assumption 3: *There are two types of information flow relevant to the challenge of location inhomogeinity: (1)* information flow of processed data *and (2)* information flow of virtual resources.
Evidence: In Section 5.1.1, three types of information flow were identified in clouds. Further, the *information flow of processed data* (cf. Def. 5.1) and that of virtual resources (cf. Def. 5.2) were identified as the most relevant for addressing the challenge of location inhomogeneity (cf. Def. 2.2), since they cover customers' data, which are sensitive to the effective level of protection. Additionally, it is assumed that the information flow of meta data does not cover customers' data and, therefore, is not investigated in this thesis. This assumption is made plausible in Section 5.1.2, due to the observation that the *information flow of processed data* changes the state of virtual resources only with respect to the data contained.

Assumption 4: *The* information flow of processed data *does not cause the* information flow of virtual resources.

Evidence: The assumption is made plausible in Section 5.1.2 by clarifying that the *information flow of processed data* changes the state of virtual resources only with respect to contained data. Further, this assumptions makes it possible to investigate the *information flow of virtual resources* separately from that of processed data.

Assumption 5: *The* information flow of processed data *is classified by categories of data and by allowed information flow.*
Evidence: This assumption is a necessary prerequisite for classifying virtual resources based on what types of data will be processed within and for controlling the *information flow of virtual resources* according to the allowed *information flow of processed data* (cf. Section 5.1.2 and *Assumption 6*). Good candidates for classifying the *information flow of processed data* are the categories (and subcategories) of data to which legal norms correspond (cf. Section 3.1.1 and Section 3.6.1). The same classification can be applied to virtual resources to control information flow.

Assumption 6: *All virtual resources are classified by categories of data, which are processed within the virtual resources. This classification does not change during the lifetime of a virtual resource, and the corporate customer utilises virtual resources according to their classification to process data of the respective category.*
Evidence: This assumption is made to avoid unwanted interference between the *information flow of processed data* and that of virtual resources and to sustain the separation of information flow control (cf. Section 5.1.2). Without this assumption, the *information flow of processed data* can result in the change of classification of virtual resources with respect to the data processed within (cf. Section 5.1.2). Such changes of classification can result in security conflicts, for example, when the new classification is forbidden for the allocated hardware resource. To prevent such conflicts, it is necessary to verify the classification of hardware resources and if necessary to migrate virtual resources to hardware resources with sufficient classification. Thus, every *information flow of processed data* can result in a reorganisation of the virtual resource placement. This implies a shift in the control of virtual resource placement from the cloud provider to the corporate customers, which is neither in the interest of corporate customers nor in the interest of cloud providers. In particular, the corporate customer has no knowledge of the classification of hardware resources and allocation to host virtual resources, and consequently, the virtual resource placement can be arbitrary and inefficient. Further, a migration has to be performed before the classification changes, i.e., before the *information flow of processed data* that implies the change of the classification. If the cloud provider prevents conflicted changes of classification then this results in the prevention of the *information flow of processed data*. Thus, this is a shift in the control of processed data from the corporate customer to the cloud provider, which is again in the interest neither of corporate customers nor of cloud providers. Consequently, it is assumed that changes of classification are not made during the lifetime of a virtual resource in order to sustain the separation of information flow control.

Assumption 7: *There exists a cloud management process* $\overline{cmp} : \mathcal{P}(\mathbb{VR}) \times \mathcal{P}(\mathbb{HW})$ *that is controlled by the cloud provider and assigns virtual resources* $\overline{VR} \in \mathcal{P}(\mathbb{VR})$ *to hardware resources* $\overline{HR} \in \mathcal{P}(\mathbb{HW})$ *(cf. Section 4.1.4.4).*
Evidence: The cloud management process was identified as a part of the cloud infrastructure

(cf. Section 4.1.4) and is defined in Section 4.1.4.4. With the cloud management process it is possible to abstract from individual types of virtual resource and hardware resources and to describe the *information flow of virtual resources* based on virtual resources and hardware resources in general.

Assumption 8: *Hardware resources are trustworthy with respect to their classification. In particular, there are no covert channels in hardware resources.*
Evidence: This assumption is made to establish the basis for reliable control of the *information flow of virtual resources* between hardware resources. Further, it helps to focus on the problem of location inhomogeneity by avoiding the modelling of information flow that occurs only on untrustworthy hardware resources (and at untrustworthy hosting sites) like the translocation of hardware resources and data extraction via physical access (both unauthorised) at the hosting sites. In practice, the trustworthiness of the hardware resources and the hardware provider is a common requirement in IT outsourcing scenarios, and is covered by IT security standards like German 'IT-Grundschutz' [30], ISO 27001 [112] and specifically by TÜViT – Trusted Site Infrastructure [206]. For the control of the *information flow of virtual resources*, virtual resources have to be assigned to hardware resources which have a classification that is suitable for the assigned virtual resource. However, the cloud providers' control of the operation of hardware resources is limited since this is in the hardware providers' sphere of responsibility. Thus, the cloud provider has to rely on the hardware provider to ensure that the hardware resources are operated according to their classification (e.g., having the same *necessary level of security*; cf. Section 3.6.1). In particular, it is important that the classification of hardware resources does not change without first notifying the cloud provider. In general, the classification of hardware only changes when it is physically manipulated (e.g., is moved to a different location or physically reconfigured), which is rather a rare event at hardware providers and is usually possible only if the hardware resource is powered off beforehand. In contrast, for mobile hardware resources, a change of location is a regular event. However, in cloud infrastructures, it is reasonable to assume that hardware resources have static locations and, moreover, are not physically manipulated by the hardware provider. But even if hardware resources have a trustworthy classification, they can be a source of unrecognised information flow. For example, virtual resources are migrated within hardware pools of virtualised data centres (and without notifying the cloud provider). Further, the hardware provider can copy or extract any information from virtual resources (e.g., by using virtual machine introspection [80]). There are methods to prevent such types of unrecognised information flow, for example, by using a trusted hypervisor [79]. Therefore, it is assumed in this thesis that hardware resources and hardware providers are trustworthy with respect to their classification and that there are no covert channels on a hardware level.

Assumption 9: *The control on the* information flow of processed data *and the* information flow of virtual resources *is decoupled.*
Evidence: *Assumption 4*, *Assumption 5*, and *Assumption 6* implies that decoupling the control of the *information flow of processed data* and that of virtual resources is possible. This is achieved by classifying both types of information flow by the categories of data that are processed by the corporate customers (cf. *Assumption 5* and *Assumption 6*). On the one hand, this implies that the *information flow of processed data* does not change the state of virtual re-

sources with respect to the processed data contained within (i.e., the classification of the virtual resource does not change). In conjunction with *Assumption 4*, this implies that the *information flow of processed data* does not interfere with the that of virtual resources. On the other hand, this implies too that the *information flow of virtual resources* is compliant with the category of data processed within. Thus, the *information flow of processed data* that is caused by the *information flow of virtual resources* is compliant with respect to the category of data, too.

Consequently, the *information flow of processed data* cannot result in a forbidden *information flow of virtual resources* and vice versa. Therefore, it is reasonable to assume that the control of the *information flow of processed data* and that of virtual resources are decoupled if *Assumption 4*, *Assumption 5*, and *Assumption 6* are satisfied.

Remark 5.3 (Focusing on information flow of virtual resources) *If the control of the* information flow of processed data *and that of virtual resources are decoupled (cf.* Assumption 9*) then the control of both types of information flow can be modelled independently. To address the challenge of location inhomogeneity (cf. Def. 4.6), it is paramount to control the* information flow of virtual resources*, since this provides a basis for reliably provisioning virtual resources with the* necessary level of security*, which is required by the corporate customers when processing data within. Therefore and without loss of generality, this thesis focuses on controlling the* information flow of virtual resources*. How the control of the* information flow of processed data *can be established on top of the control of the* information flow of virtual resources *is discussed in Section 7.3.*

5.2 Limits of existing models for information flow control

In Section 5.1, the need to control the information flow was identified. Existing models for information flow control aim to achieve access control and propagation/transmission control of *objects* containing information with restrictions on access to them and their propagation. Entities which are accessing objects (via retrieval or propagation) are considered *subjects*. Further, objects and subjects are classified by access and propagation permissions (e.g., clearance and need-to-know levels [20]). This makes it possible to define relations (i.e., rules) between the different classes of objects and subjects describing the allowed information flow between them.

This section identifies the methods for modelling information flow control, which are applicable to tackling the challenge of location inhomogeneity (cf. Def. 4.6). For that purpose, existing models are analysed and the applicable models which are used in this thesis, are introduced. As a result, a lattice-based model for access control based on existing models for confidentiality and integrity is presented. Further, the design requirements are formulated for a generalised model addressing the challenge of location inhomogeneity in clouds.

The remainder of this section is organised as follows. First, suitable models for information flow control that apply to the scenario of IT outsourcing are identified. Then, the relevant models are introduced. Finally, the limits of their application in clouds for addressing the challenge of location inhomogeneity are discussed.

5.2.1 Mandatory access control vs. discretionary access control

Models for information flow control are distinguished by Mandatory Access Control (MAC) and Discretionary Access Control (DAC). MAC describes models with a centralised hierarchy of classifications with rules that allow the definition of access decisions based on the classification of subjects and objects. DAC describes models with a decentralised hierarchy of actor-centric rules which define access to objects that belong to specific subjects (i.e., subjects own objects and control these objects) [185] [131]. In this section the two concepts are analysed to determine which of the two is suitable for addressing the challenge of location inhomogeneity (cf. Def. 4.6).

To address the challenge of location inhomogeneity, it is necessary to control the *information flow of virtual resources* (cf. Remark 5.3). This is controlled by the cloud providers, since they operate the cloud management process assigning virtual resources to hardware resources (cf. Section 5.1.2). The subjects are then the hardware resources and objects the virtual resources. There exist two possible cases: (1) single-cloud scenarios having only a single cloud provider involved and (2) multi-cloud scenarios where multiple cloud providers are involved.

In single-cloud scenarios, the cloud provider acts as the central authority controlling access to virtual resources with respect to utilisation by corporate customers and placement on hardware resources. Thus, access control can be established by a centralised hierarchy of virtual resource classification that is controlled by the cloud management process, which is MAC. Further, DAC does not fit very well in the single-cloud scenario since there are is only a single authority implementing the rules.

In multi-cloud scenarios, there are multiple cloud providers involved. Each virtual resource can be associated with the cloud provider that is responsible for it, and therefore, there is an owner relationship between each virtual resource and its corresponding cloud provider. Further, virtual resources are classified based on the requirements of their owner, i.e., the requirements requested by the utilising corporate customers which contracted the respective cloud provider. However, the ownership of virtual resources is not a sufficient requirement when implementing information flow control with respect to virtual resource placement on hardware resources. It is also necessary that hardware resources have a sufficient classification, which is independent of the responsible cloud provider. In particular, cloud providers usually have no knowledge of the hardware resources and their classification used by other cloud providers, since they only utilise the virtual resources of other cloud providers (similar to corporate customers). Therefore, a centralised hierarchy of classification is required that is applicable cross-platform when controlling the *information flow of virtual resources*. In particular, a cloud provider should not grant access to virtual resources to another cloud provider if the other does not have hardware resources with sufficient classification. The latter is difficult to address in DAC, since there is by design no central authority that enforces rules on information flows across multiple subjects. In particular, there is no guarantee of the transitivity of information flow control, which is important to enforce *transfer control* (cf. 3.6.3.3). This can be exemplified by an information flow between three subjects S_1, S_2, S_3. If information is allowed to flow from S_1 to S_2 and from S_2 to S_3 then (due to the transitivity of information flow) information can flow from S_1 to S_3. However, if S_1 does not allow information flow to S_3 then this rule is undermined by the transitive information flow via S_2. To solve this, a centralised hierarchy of classification

is required. Therefore, MAC is a good candidate for information flow control in clouds, since it supports the control of transitive information flow (while DAC does not).

To conclude, MAC is more suitable than DAC for both single- and multi-cloud scenarios. However, in practice, it is difficult to introduce a central authority in multi-cloud scenarios enforcing the information flow control between different clouds. This issue can be addressed using the decentralised approach of MAC with a central hierarchy of classification but decentralised authorities defining rules on accessing the objects they own. An example of such an approach is the definition of (1) security labels that are attached to objects specifying their classification and (2) rules on how security labels are combined if objects belonging to different subjects are merged [146].

In this thesis, information flow control is modelled using decentralised MAC methodes from the perspective of a single cloud provider. Without loss of generality, other cloud providers are treated like corporate customers when utilising virtual resources and like hardware providers with multiple classifications of hardware resources when hosting virtual resources.

5.2.2 Lattice-based models for access control

To address the challenge of location inhomogeneity (cf. Def. 4.6), it is necessary to control information flow with respect to confidentiality, integrity, availability, and location determination (cf. *Objective 5*; Section 1.3). In Section 5.2.1, decentralised MAC was identified as a good candidate for establishing information flow control in clouds. Existing work in MAC already addresses confidentiality [20] [52] and integrity [22]. However, none of these models addresses availability and location constraints. There is research on information flow control addressing the co-location of virtual resources [201] and user location relative to resource location [166] (cf. Section 2.4). None of these approaches are capable of dealing with location constraints of hardware resources directly.

In this section, existing models for information flow control with respect to confidentiality and integrity that are adopted in this thesis are introduced. Further, it is shown that the models of Bell and La Padula [20], Denning [52], and Biba [22] are compatible, and therefore can be combined into a single lattice-based model for access control. Afterwards, existing models for information flow control are discussed with respect to issues on tackling the challenge of location inhomogeneity. In particular, the need for information flow control with respect to availability and location determination is investigated.

5.2.2.1 Confidentiality-based information flow control

Confidentiality-based information flow control was introduced by the fundamental work of Bell and La Padula [20][127]. In their approach, information flow control is modelled by defining access privileges for reading and writing on objects, classifying subjects and objects, and based on that, defining rules on deciding whether a subject is allowed to have an access privilege. By using a mathematical model, it is possible to prove whether a system with specific rules is secure in the sense of confidentiality. For this purpose, secure information flow is addressed on two levels: (1) information flow between subjects and objects (a.k.a. *simple-security property*)

and (2) information flow between objects (a.k.a. $*$-*property*). In the following, the model of Bell and La Padula is introduced with definitions of its terms and concepts.

Model definition (according to Bell and La Padula [20][127])
First, the entities are defined in terms of where information can flow. These entities are distinguished by whether they are active or passive.

Definition 5.3 (Subject [127]) *A* subject *is defined as an active entity interacting with other entities (e.g., processes, programs in execution). Then,* $\mathbb{S} := \{S_1, ..., S_n\}$ *is the set of subjects* S_i *and* $\mathbb{S}^+ := \mathbb{S} \cup \emptyset$.

Definition 5.4 (Object [127]) *An* object *is defined as a passive entity with which subjects interact with (e.g., data, files, programs). Then,* $\mathbb{O} := \{O_1, ..., O_n\}$ *is the set of subjects* O_i.

For controlling information flow between subjects and objects, both are classified in two dimensions: by clearance level and by need-to-know level. These two dimensions derive from the access policies used in the military context which the model was originally designed for. A two-dimensional classification makes it possible to define a general hierarchy for classification (i.e., clearance level) and to refine this hierarchy by introducing additional classifications that are limited to a specific context (i.e., need-to-know level). Examples of classification by specific context are information flow control based on projects and divisions of a company.

Definition 5.5 (Classification [127]) *A* classification *is defined as a clearance level of subjects and objects. Classifications are ordered by the* $>$*-relation. Then,* $\mathbb{C} := \{C_1, ..., C_n\}$ *is the set of classifications* C_i *where* $C_1 > ... > C_n$.

Definition 5.6 (Category [127]) *A* category *is defined as a need-to-know level of subjects and objects and expresses a special access privilege provided in a specific context (e.g., project, division of a company). Then* $\mathbb{K} := \{K_1, ..., K_n\}$ *is the set of categories* K_i.

To map classifications and categories to subjects and objects, a relation is defined which is denoted *classification/need-to-knows vector*.

Definition 5.7 (Classification/needs-to-know vector [127]) *The* classification/needs-to-know vector *is defined as the relation* $\overline{F}$: $\mathbb{C}^{\mathbb{S}} \times \mathbb{C}^{\mathbb{O}} \times \mathcal{P}(\mathbb{K})^{\mathbb{S}} \times \mathcal{P}(\mathbb{K})^{\mathbb{O}}$, *where* $\overline{f} \in \overline{F}$ *is defined* $\overline{f} := (\overline{f_1}, \overline{f_2}, \overline{f_3}, \overline{f_4})$ *with*

- $\overline{f_1} : \mathbb{S} \to \mathbb{C}$, *subject-classification function, where each subject is mapped to a classification;*
- $\overline{f_2} : \mathbb{O} \to \mathbb{C}$, *object-classification function, where each object is mapped to a classification;*
- $\overline{f_3} : \mathbb{S} \to \mathcal{P}(\mathbb{K})$, *subject-category function, where each subject is mapped to a set of categories; and*

- $\overline{f_4} : \mathbb{S} \rightarrow \mathcal{P}(\mathbb{K})$, *object-category function, where each object is mapped to a set of categories.*

Further, a specification determines how a subject is allowed to access an object. Therefore, access is distinguished by whether there is information flow and by the direction of information flow. For example, there is no information flow if the subjects changes the classification of an object, since no information flows between subject and object. The same is true if the subject executes the object, i.e., the subject triggers actions of the object (e.g., a program is started). The direction of information flow is distinguished by: (1) unidirectional from object to subject (view), (2), unidirectional from subject to object (modify), and (3) bidirectional (view and modify).

Definition 5.8 (Access attributes [127]) $\mathbb{A} := \{\underline{r}, \underline{a}, \underline{w}, \underline{e}, \underline{c}\}$ *is defined as the set of access attributes where the access attributes are defined as follows.*

- $\underline{r}$ *(read): subjects are allowed to view objects*
- $\underline{a}$ *(append): subjects are allowed to modify objects*[1]
- $\underline{w}$ *(write): subjects are allowed to view and modify objects*
- $\underline{e}$ *(execute): subjects are allowed to execute objects*
- $\underline{c}$ *(control): subjects are allowed to extend access to other subjects*

To assign access attributes to objects with respect to specific subjects, a matrix is defined that represents the current configuration of access attributes for each object with respect to all subjects. Considering that there are 2^5 combinations of access attributes for each entry, the number of all possible access matrices is $c := |S| \cdot |O| \cdot 2^5$.

Definition 5.9 (Access matrix [127]) *An access matrix is defined as a $|S|x|O|$-matrix with entries from $\mathcal{P}(\mathbb{A})$, where the entry at (i,j) shows the access attributes of subject S_i relative to object O_j. Then, $\mathbb{M} := \{M_1, ..., M_c\}$ with $c = |S| \cdot |O| \cdot 2^5$ is the set of all possible access matrices M_i.*

To model systems that control information flow between subjects and objects they are defined in a similar way to state machines, with access requests as input and control decisions as output. In each state, there are subjects having access to objects in specific access modes, and there is an access matrix and a classification/needs-to-know vector which define allowed information flows. Due to a state transition, each of these properties may change. For example, if a subject S requests access to an object O in read mode (i.e., access privilege $\underline{r}$) and this request is allowed then $(S, A, \underline{r})$ is added to the next state. Further, multiple subsequent state-transitions are described as a sequence of subsequent states.

[1] In this thesis, the original definition according to Bell and La Padula [20] is used including override and deletion of unknown content, since this is the most general case of modifying an object. For practical reasons, append can be implemented by only adding information to an object but not overriding and deleting existing information (for instance making changes to the end of an file in file systems). For information flow control both cases are valid object modifications and no information is read from the object.

Definition 5.10 (State [127]) *A state V is defined as a triple $V := (b, M, f)$ with*

- *$b \subseteq \mathbb{S} \times \mathbb{O} \times \mathbb{A}$ set of all subjects $S \in \mathbb{S}$ having access to objects $O \in \mathbb{O}$ in what access mode, which is described by a set of access attributes $A \subseteq \mathbb{A}$;*
- *$M \in \mathbb{M}$ access matrix in the state V; and*
- *$f \in \overline{F}$ classification/needs-to-know vector describing classification of all subjects and objects and the categories associated with the subjects and objects.*

Then, $\mathbb{V}$is the set of states V_i.

Definition 5.11 (State sequence [127]) *A state sequence is an arbitrary number of chronologically ordered states $V_i \in \mathbb{V}$. Then, $\mathbb{Z} : \mathbb{V}^{\mathbb{N}}$ is the set of state sequences Z_i.*

Each state transition is triggered by an access request of a subject. There are seven different types of request possible: (1) *get* access to an object, (2) *release* access to an object, (3) *give* access privileges to an other subject, (4) *rescind* access privileges of an other subject, (5) *change* the classification/category of an object, (6) *create* an object, and (7) *delete* an object. This implies that each request can contain information on up to two subjects, on an object, on access attributes, and on the classification/needs-to-know vector. Further, the requests of multiple subsequent state-transitions are described as a sequence of subsequent requests.

Definition 5.12 (Request [127]) *A request is defined as a quadruple $(S_1, S_2, O, G) \subset \mathbb{S}^+ \times \mathbb{S}^+ \times \mathbb{O} \times \mathbb{G}$ with $\mathbb{G} := \mathbb{A} \cup \emptyset \cup \overline{F}$. Then, $\mathbb{R} : \mathbb{S}^+ \times \mathbb{S}^+ \times \mathbb{O} \times \mathbb{G}$ is the set of requests R_i.*

Definition 5.13 (Request sequence [127]) *A request sequence is an arbitrary number of timely ordered requests $R_i \in \mathbb{R}$. Then, $\mathbb{X} : \mathbb{R}^{\mathbb{N}}$ is the set of request sequences X_i.*

Definition 5.14 (Request elements [127]) Request elements *are requests by a subject. There exist seven requests, where*

- *'get' is the request to gain access to an object;*
- *'give' is the request to give access to another subject with respect to some objects;*
- *'release' is the request to remove access to an object;*
- *'rescind' is the request to remove access of another subject with respect to some objects;*
- *'change' is the request to change the classification/needs-to-know vector for some objects;*
- *'create' is the request to create an object in the system; and*
- *'delete' is the request to delete an object from the system.*

In response to a request, the system returns a decision which is applied to the current state resulting in a state-transition. While the answer to a request usually is that it is either allowed or forbidden, it is possible that there is no clear decision, which is considered a conflict. In particular, this is the case if more than one decision is possible. Further, it is possible that the request is not recognised by the system, for example, an unknown access attribute is used in the request. Again, the decisions of multiple subsequent state-transitions are described as a sequence of subsequent decisions.

Definition 5.15 (Decisions [127]) *A decision is defined as a response to be given to any request. The set of decisions is defined* $\mathbb{D} := \{\underline{yes}, \underline{no}, \underline{error}, \underline{?}\}$ *where the decisions are defined as follows.*

- $\underline{yes}$*: The request is allowed.*
- $\underline{no}$*: The request is denied.*
- $\underline{error}$*: The request cannot be decided due to a conflict.*
- $\underline{?}$*: The request is not recognised and no decision is made.*

Definition 5.16 (Decision sequence [127]) *A decision sequence is an arbitrary number of chronologically ordered decisions* $D_i \in \mathbb{D}$*, and* $\mathbb{Y} : \mathbb{D}^{\mathbb{N}}$ *is the set of decision sequences* Y_i*.*

Having defined requests, decisions and states, a system can be described by a request sequence, the resulting decision sequence, the corresponding state-transitions, and its initial state. In particular, each system has a *state transition relation* describing for each request and each state the corresponding decision and next state.

Definition 5.17 (System [127]) *Let* $\overline{W} \subset \mathbb{R} \times \mathbb{D} \times \mathbb{V} \times \mathbb{V}$*. The system* $\Sigma(\mathbb{R}, \mathbb{D}, \overline{W}, z_0) \subset \mathbb{X} \times \mathbb{Y} \times \mathbb{Z}$ *is defined by* $(X, Y, Z) \in \Sigma(\mathbb{R}, \mathbb{D}, \overline{W}, z_0)$ *if and only if* $(X_t, Y_t, Z_t, Z_{t-1}) \in \overline{W}$ *for each* $t \in \mathbb{N}$ *where* $z_0 := (\emptyset, M, f)$ *is initial state with* $M \in \mathbb{M}$ *initial access matrix and* $f \in \overline{F}$ *initial classification/needs-to-know vector.* $\overline{W}$ *is denoted* state transition relation.

A system is considered secure if subjects only have access to objects according to the classification and categories. With respect to confidentiality, a subject should gain access to an object only if information is allowed to flow from the object to the subject and vice versa. Information can flow from an object to a subject only if the subject can view the object, i.e., accessing in *read*-mode $\underline{r}$ or *write*-mode $\underline{w}$. On the other hand, information can flow from a subject to an object, only if the subject can modify the object, i.e., accessing in *append*-mode $\underline{a}$ or *write*-mode $\underline{w}$. Information flow from objects to subjects is addressed by the *simple-security property*, while information flow from subjects to objects is addressed by the $*$-property.

Definition 5.18 (Simple-security property [127]) $(S, O, A) \in \mathbb{S} \times \mathbb{O} \times \mathbb{A}$ *satisfies the security condition relative to* $f = (f1, f2, f3, f4) \in \overline{F}$ *if and only if*

(i) $(A = \underline{e}) \vee (A = \underline{a}) \vee (A = \underline{c})$

(ii) $((A = \underline{r}) \vee (A = \underline{w})) \wedge ((f_1(S) \geq f_2(O)) \wedge (f_3(S) \supseteq f_4(O)))$.

Further, secure *and* compromise *can be defined for states, state sequences, appearance of a system and a system.*

A state $V = (b, M, f) \in \mathbb{V}$ *is a* secure state *if and only if each* $(S, O, X) \in b$ *satisfies the security condition relative to* f*. Otherwise, the state* v *is a* compromise state.

A state sequence $Z \in \mathbb{Z}$ *is a secure state sequence if and only if every state* $Z_t \in Z$ *is a secure state. Otherwise, the state sequence* Z *has a compromise.*

An appearance $(X, Y, Z) \in \Sigma(\mathbb{R}, \mathbb{D}, \overline{W}, z_0)$ *of the system* $\Sigma(\mathbb{R}, \mathbb{D}, \overline{W}, z_0)$ *is a secure appearance if and only if* Z *is a secure state sequence. Otherwise, the appearance* (X, Y, Z) *has a compromise.*

The system $\Sigma(\mathbb{R}, \mathbb{D}, \overline{W}, z_0)$ *is secure if and only if every appearance of* $\Sigma(\mathbb{R}, \mathbb{D}, \overline{W}, z_0)$ *is secure. Otherwise, the system* $\Sigma(\mathbb{R}, \mathbb{D}, \overline{W}, z_0)$ *has a compromise.*

In addition to the control of information flow from subjects to objects, it is also necessary to control information flow between objects. In particular, there sholud not be information flow between objects not having the sufficient classification and categories. Information flow between objects is only possible if a subjects views an object and then modifies another. Therefore, information flow from a subject to an object (i.e., the subject modifies the object) is allowed only if (1) the subject has sufficient classification and categories, and (2) all objects viewed by the subject have no classification or categories that are not covered by the modified object. Both are addressed by the $*$-property.

Definition 5.19 ($*$-property [127]) *Let* $b(S : A_1, ..., A_k)$ *the set of objects where subject* $S \in \mathbb{S}$ *has access in the access modes* $A_1, ..., A_k \in \mathbb{A}$ *and* $k \in \mathbb{N}$ *for a given state* $V = (b, M, f)$*, i.e.,*

$$b(S : A_1, ..., A_k) := \{O : O \in \mathbb{O} \wedge (\forall i \in \{1, ..., k\} : (S, O, A_i) \in b)\}$$

where b *is the set of all subjects having access to which objects in which access mode of the respective state,* $M \in \mathbb{M}$ *is access matrix, and* $f \in \overline{F}$ *is classification/needs-to-know vector (cf. Def. 5.10).*

A state $V = (b, M, f) \in \mathbb{V}$ *satisfies the* $*$*-property if and only if*

$$\begin{aligned} &\forall S \in \mathbb{S} : b(s : \underline{w}, \underline{a}) \neq \emptyset \wedge b(s : r, w) \neq \emptyset \\ &\quad \Rightarrow \forall O_1 \in b(s : w, a), O_2 \in b(s : r, w) : f_2(O_1) \geq f_2(O_2) \wedge f_4(O_1) \supseteq f_4(O_2) \,. \end{aligned}$$

Analogously to (simple-security) secure condition (cf. Def. 5.18), the satisfaction of the $*$*-property is defined for state sequences, appearances of a system, and system* $\Sigma(\mathbb{R}, \mathbb{D}, \overline{W}, z_0)$*.*

Based on the simple-security property and the $*$-property, it is possible to design secure systems that satisfy the $*$-property. Therefore, it is necessary to define rules that map requests to decisions according to the current state of the system and return the next state based on the decision. In a secure system, only secure states are allowed. Therefore, each rule has to be *security preserving*, which means that it maps secure states to secure next states only. Analogously, in systems that satisfy the $*$-property, each rule has to be $*$-property preserving.

Definition 5.20 (Rule [127]) *A rule is a function* $\rho : \mathbb{R} \times \mathbb{V} \to \mathbb{D} \times \mathbb{V}$ *that maps a request and a state to a decision and a state, analogously to a state-transition function of a finite-state machine.*

A rule ρ *is* security persevering *if and only if*

$$\forall (R,V) \in \mathbb{R} \times \mathbb{V}\ \exists D \in \mathbb{D}\ \exists V' \in \mathbb{V}:$$
$$\rho(R,V) = (D,V') \wedge V \textit{ is secure state} \Rightarrow V' \textit{ is secure state.}$$

Analogously, a rule ρ *is* $*$-property preserving *if and only if state V satisfies* $*$*-property implies state* V' *satisfies the* $*$*-property.*

If a rule handles a request then decision $D \in \{\underline{yes}, \underline{no}, \underline{error}\}$*, otherwise* $D = \underline{?}$.

Given a set of rules. The system response D to a request R is defined:

- $D = \underline{?}$ *if no rule handles the request;*
- $D = \underline{error}$ *if more than one rule is applicable;*
- $D = \underline{yes}$ *if a unique rule is applicable and the decision of the rule is* $\underline{yes}$*; and*
- $D = \underline{no}$ *if a unique rule is applicable and the decision of the rule is* $\underline{no}$.

Based on the request elements (cf. Def. 5.14), LaPadula et al. defines $\Omega := \{\rho_1, ..., \rho_{10}\}$ a set of rules [127] for a secure system that satisfy the $*$-property. The rules $\rho_1, ..., \rho_{10}$ can be summarised as follows.

- **Rule 1 (get-read)** ρ_1**:** A subject S gets read access to an object O if
  - (i) (security preserving) the access attribute $\underline{r}$ is an element of the corresponding entry of the access matrix, S has the same as or higher classification than O and the categories of S cover the categories of O; and
  - (ii) ($*$-property preserving) all objects O' where S can write to (i.e., access in append or write mode) have the same as or higher classification than O and their categories cover the categories of O.
- **Rule 2 (get-append)** ρ_2**:** A subject S gets append access to an object O if
  - (i) (security preserving) the access attribute $\underline{a}$ is as element of the corresponding entry of the access matrix; and
  - (ii) ($*$-property preserving) all objects O' that S can read from (i.e., access in read or write mode) have a lower classification than O and their categories are covered by the categories of O.
- **Rule 3 (get-execute)** ρ_3**:** A subject S gets execute access to an object O if
  - (i) (security preserving/$*$-property preserving) the access attribute $\underline{e}$ is element of the corresponding entry of the access matrix.
- **Rule 4 (get-write)** ρ_4**:** A subject S gets execute access to an object O if
  - (i) (security preserving) the access attribute $\underline{w}$ is an element of the corresponding entry of the access matrix, S has the same as or higher classification than O and the categories of S cover the categories of O; and

(ii) ($*$-property preserving [append]) all objects O' where S has append access have the same as or higher classification than O and their categories cover the categories of O.

(iii) ($*$-property preserving [read]) all objects O' where S has read access have lower classification than O and their categories are covered by the categories of O.

(iv) ($*$-property preserving [write]) all objects O' where S has write access have the same classification as O and their categories are the same with that of O.

- **Rule 5 (release-read/write/append/execute)** ρ_5**:** A subject S can release read/write/append/execute access to an object O (if the request is valid).

- **Rule 6 (give-read/write/append/execute)** ρ_6**:** A subject S gives to a subject S' read-/write/append/execute access to an object O if

  (i) (security preserving/$*$-property preserving) S has read/write/append/execute access and control access to O.

- **Rule 7 (rescind-read/write/append/execute)** ρ_7**:** A subject S rescinds read/write/append/execute access to an object O for a subject S' if

  (i) (security preserving/$*$-property preserving) S has read/write/append/execute access and control access to O.

- **Rule 8 (change-f)** ρ_8**:** A subject S can change the classification/need-to-know vector f if

  (i) (security preserving/$*$-property preserving) S changes only classifications and categories of objects that no subject has access to.

- **Rule 9 (create-object)** ρ_9**:** A subject S can create an object O if

  (i) (security preserving/$*$-property preserving) O does not exist.

- **Rule 10 (delete-object)** ρ_{10}**:** A subject S can delete an object O if

  (i) (security preserving/$*$-property preserving) the access attribute $\underline{c}$ is an element of the corresponding entry of the access matrix.

Based on the rules Ω, LaPadula et al. formulates the basic Security Theorem for secure systems which satisfy the $*$-property:

Theorem 5.1 (Security theorem [127]) *The rules Ω are security preserving and $*$-property preserving. Further, a system $\Sigma(\mathbb{R}, \mathbb{D}, \overline{W}, z_0)$ using theses rules is a secure system and satisfies the $*$-property if Z_0 is a secure state which satisfies the $*$-property.*

Proof The proof of the *Security Theorem* is provided by LaPadula et al. [127]. ■

Further, LaPadula et al. identifies two properties that are important for the design of rules in secure systems:

Definition 5.21 (Covering and disjoint [127]) *A set of rules is considered* covering *if there is always an applicable rule. Further, a set of rules is considered* disjoint *if there is only a single rule applicable (i.e., for all requests there is no error in response).*

Covering is important, because otherwise there are requests that remain unrecognised by the system. Thus, it is not possible to distinguish between unrecognised requests that were forgotten in the design of rules and unrecognised requests that are illegal. In particular, when testing a system, it is important to identify illegal and forgotten requests clearly.

Disjoint is important for avoiding conflicts which are considered a flaw in the design of rules. In particular, a conflict indicates that there exist requests that are not modelled correctly in the design of the rules.

Remark 5.4 (Covering and disjoint Ω [127]) *It can be observed that the rules in Ω are disjoint but not covering. However, covering can be achieved by introducing a new decision, called illegal, and a 'catching rule' that applies if no other rule apply. Then, when ever the 'catching rule' applies, the decision illegal is returned indicating that the request is not processed by the system.*

5.2.2.2 Lattice-based information flow control

The model of Bell and La Padula focuses on direct access control of subjects and objects based on a two-dimensional classification of confidentiality. However, the approach lacks of the ability to model access control based on information flow in general. A more general approach to modelling information flow control was presented by Denning [52]. In Denning's approach, a lattice of so called *security classes* is used for classification of subjects and objects, and access is controlled based on the order in the lattice and with respect to information flow between security classes. In comparison to Bell and La Padula, decisions on access control have a reduced complexity since they are based on information flow between security classes organised in a lattice. This reduces the decision to comparing two security classes. Additionally, the model is more flexible with respect to the expressiveness of the classification and its management. In particular, there are rules for how security classes are combined. This flexibility is also key for extending the model to cover multiple security properties (in this thesis: confidentiality, integrity, availability, and location determination) which are exploited in Section 5.3 to develop a model on information flow control in clouds. Further, the simplicity and flexibility of Denning's approach ensures that it is compatible with Bell and La Padula's approach, and as a result, both models can be combined into a single model which is demonstrated in this section.

In the following, Denning's model on lattice-based access control [52] is introduced, and based on the model of Bell and La Padula, both models are combined into a lattice-based model for information flow control.

Model definition (according to Denning [52])
Analogously to the model of Bell and La Padula (cf. Section 5.2.2.1), there exist *subjects* accessing *objects*. Hence, the definition of Bell and La Padula is applicable, too (cf. Def. 5.3 and Def. 5.4). Subjects and objects are classified by *security classes*, which cover classification (cf. Def. 5.5) and categories (cf. Def. 5.6) of Bell and La Padula's model but are not limited to them. In the Denning's model, access control decisions are made on the basis of allowed information flow between security classes. For that purpose, two compare relations are defined. The first defines how security classes are allowed to be combined, which is needed to model the join of information having different security classes. Also, it can be used to identify the security class that is necessary to access multiple objects associated with different security classes. The second defines the allowed information flow between security classes which is required to determine whether a subject is allowed to access an object. Having this in mind, the model is defined as follows.

Definition 5.22 (Information flow model [52]) *An information flow model is defined as a quintuple* $(\mathbb{S}, \mathbb{O}, \mathbb{SC}, \oplus, \mapsto)$ *where*

- $\mathbb{S}$ *is a set of subjects (cf. Def. 5.3);*
- $\mathbb{O}$ *is a set of objects (cf. Def. 5.4);*
- $\mathbb{SC} := \{SC_1, ..., SC_n\}$*, a set of security classes* SC_i*;*
- $\mathbb{SCB} : (\mathbb{S} \cup \mathbb{O}) \times \mathbb{SC}$ *is a set of security bindings where each element is a binding of an object or subject to a security class*
- $\oplus : \mathbb{SCB} \times \mathbb{SCB} \to \mathbb{SCB}$ *with* $\oplus(SC_1, SC_2) = SC_1 \oplus SC_2 = SC_3$ *and* $SC_1, SC_2, SC_3 \in \mathbb{SCB}$ *is a class-combining operator that provides for a pair of security bindings* (SC_1, SC_2) *a combined security binding* SC_3*; and*
- $\mapsto$: $\mathbb{SC} \times \mathbb{SC}$ *is a flow relation describing an allowed information flow between two security classes with* $SC_1 \mapsto SC_2$ *if and only if information in* SC_1 *is allowed to flow to* SC_2 *(*$SC_1, SC_2 \in \mathbb{SC}$*).*

After defining the model, the lattice on security classes is established. For that reason, two compare relations are required. The flow relation and the class-combining operator are compare relations like those proved by Denning [52]. In particular, Denning formulated four axioms that have to be satisfied to establish a lattice as follows.

Theorem 5.2 (Denning's Axioms [52]) $(\mathbb{SC}, \mapsto, \oplus)$ *is a lattice, since the following assumptions (*Denning's Axioms*) hold true.*

(1) $(\mathbb{SC}, \mapsto)$ *is a partially ordered set.*

(2) $\mathbb{SC}$ *is finite.*

(3) $\mathbb{SC}$ *has a lower bound* SC_L *such that* $SC_L \mapsto SC$ *for all* $SC \in \mathbb{SC}$*.*

(4) $\oplus$ *is a least upper bound operator on* $\mathbb{SC}$.

Further, these assumptions imply that there exists a greatest upper bound operator*, which is denoted by* $\otimes$*.Additionally, the existence of* $\otimes$ *implies that there exists an* upper bound SC_H *such that* $SC \mapsto SC_H$ *for all* $SC \in \mathbb{SC}$*.From this follows that* $(\mathbb{SC}, \mapsto, \oplus, \otimes)$ *is a lattice with upper bound* SC_L *and lower bound* SC_H.

Proof The proof of *Denning's Axioms* is provided by Denning [52]. ■

Remark 5.5 (Modifications on the model) *Any modifications on the model that are made in this thesis have to satisfy these axioms to ensure that the security classes still form a lattice and that all statements made on the basis of these axioms still apply.*

5.2.2.3 Joint model on lattice-based information flow control for confidentiality

A lattice-based approach for modelling information flow between security classes provides an elaborate but simple model for deciding on access control. However, it lacks the ability to distinguish between the different modes of access a subject can have (i.e., read, write, append, execute, control; cf. and Def. 5.8). It is possible to model for each mode of access the corresponding information flow – for example, subject read object is modelled as information flows from object to subject – but it is rather complex to introduce specific access modes for each possible pair of subject and object (which is done in the model of Bell and La Padula). In particular, changing allowed access modes would require a change of associated security classes and may even require a change of the flow relation. To overcome this flaw, it is necessary to combine lattice-based security classes and access modes in a single model.

An obvious approach is to combine the models of Bell and La Padula and Denning. This can be achieved by a four-step modification of the model of Bell and La Padula: In the first step, it is necessary to define a unified classification. Therefore, classifications and category are mapped on security classes to enable the utilisation of security classes for access control decisions. Then, as a second step, it is possible to bind subjects and objects to security classes as a replacement for classification/needs-to-know vector. In a third step, it is necessary to redefine the simple security property and the $*$-property which now have to be applicable on security classes instead of classifications and categories. In the fourth step, the rules have to be adjusted in such a way that the decisions are made on security classes.

Based on these steps, a lemma on constructing a combined model is formulated as follows.

Lemma 5.1 (Lattice-based information flow control) *The model for information flow control by Bell and La Padula (cf. Section 5.2.2.1) and the information flow model by Denning (cf. Section 5.2.2.2) are compatible and can be combined into a single model by:*

(1) mapping classifications in $\mathbb{C}$ *and categories in* $\mathbb{K}$ *to security classes in* $\mathbb{SC}$*, which are denoted* confidentiality classes*;*

(2) using security bindings $\mathbb{SCB}$ *(cf. Def. 5.22) instead of the classification/needs-to-know vector* $\overline{F}$ *(cf. Def. 5.7);*

(3) redefining the simple-security property (cf. Def. 5.18) and the $*$-property (cf. Def. 5.19) based on the mapped security classes (instead of classifications and categories); and

(4) redefining Ω based on mapped security classes (instead of classifications and categories). The redefined set of rules is denoted ${}^{c}\Omega$.

Proof To prove Lemma 5.1, it is shown that (i) a combined model exists (by defining one), (ii) Denning's Axioms (cf. Theorem 5.2) are satisfied by the combined model, and (iii) the security theorem (cf. Theorem 5.1) is shown for the combined model).

Proof of part (i): Construction of the combined model (proof of existence) It is shown in Appendix B that it is possible to define a c-system ${}^{c}\Sigma({}^{c}\mathbb{R},\mathbb{D},{}^{c}\overline{W},{}^{c}z_0)$ with a set of c-rules ${}^{c}\Omega$ which is analogous to the system $\Sigma(\mathbb{R},\mathbb{D},\overline{W},z_0)$ with a set of rules Ω (cf. Section 5.2.2.1) and which uses the set of confidentiality classes ${}^{c}\mathbb{SC}$ being a subset of the set of security classes $\mathbb{SC}$ (cf. Section 5.2.2.2).

The allowed information flow between confidentiality classes is defined on the bases of the comparison of classifications and categories, which are mapped to the respective confidentiality classes. Then, confidentiality classes are defined as follows.

Definition 5.23 (Confidentiality class) *A confidentiality class ${}^{c}SC := (C,K)$ is a security class with $C \in \mathbb{C}$ and $K \in \mathbb{K}$. Then, ${}^{c}\mathbb{SC} := \{{}^{c}SC_1,\ldots,{}^{c}SC_n\}$ is set of confidentiality classes ${}^{c}SC_i$ and ${}^{c}\mathbb{SC} \subseteq \mathbb{SC}$.*

*Further, let ${}^{C}SC_1 \mapsto {}^{c}SC_2$ for all ${}^{C}SC_1, {}^{C}SC_2 \in {}^{c}\mathbb{SC}$ with ${}^{c}SC_1 = (C_1,K_1)$, ${}^{c}SC_2 = (C_2,K_2)$, and $C_1,C_2 \in \mathbb{C}$ and $K_1,K_2 \in \mathbb{K}$ **if and only if** $C_1 \leq C_2 \wedge K_1 \subseteq K_2$.*

Further, the *simple-security property* (cf. Def. 5.18) is redefined on information flow between confidentiality classes. The original *simple-security property* addresses the information flow from objects to subjects, which is the case when a subject accesses an object in *read*-mode or *write*-mode. Accessing an object in *read*-mode or *write*-mode is allowed only if the subject's classification is greater than the object's classification and the object's categories are contained in the subject's categories (cf. Def. 5.18). According to the definition of confidentiality classes (cf. Def.5.23), this condition is equivalent to the condition that information is allowed to flow from the object to the subject. This equivalence is used to redefine the *simple-security property* on information flow between confidentiality classes. The new property is denoted the *simple-confidentiality property* and defined as follows.

Definition 5.24 (Simple-confidentiality property) *$(S,O,A) \in \mathbb{S} \times \mathbb{O} \times \mathbb{A}$ satisfies the security confidentiality condition relative to ${}^{c}SCB \subseteq {}^{c}\mathbb{SCB}$ if and only if*

(i) $(A = \underline{e}) \vee (A = \underline{a}) \vee (A = \underline{c})$*; and*

(ii) $((A = \underline{r}) \vee (A = \underline{w})) \wedge ({}^{c}\overline{scb}(O) \mapsto {}^{c}\overline{scb}(S))$

with ${}^{c}\overline{scb}(S) \in {}^{c}SCB$ security binding of subject S and ${}^{c}\overline{scb}(O) \in {}^{c}SCB$ security binding of object O.

Analogously to the simple-security property (cf. Def. 5.18), secure *and* compromise *are defined for c-states, c-state sequences, appearances of a c-system and a c-system.*

To redefine the $*$-*property* (cf. Def. 5.18) on information flow between confidentiality classes, the conditions on sufficient classification and categories of the accessed objects are replaced by the equivalent condition on allowed information flows between objects' confidentiality classes. The $*$-*property* addresses the information flow from subjects to objects and between objects (caused by a subject). Both are implicated if subjects access objects in *append*-mode or *write*-mode. Accessing an object in *append*-mode or *write*-mode is allowed only if all objects that can be accessed by the subject in *read*-mode or *write*-mode have a classification that is less or equal to the accessed object and each of them has a category that contains the accessed object's category (cf. Def. 5.19). This condition is equivalent to the condition that information is allowed to flow from each object that can be accessed by the subject in *read*-mode or *write*-mode to the accessed object (cf. Def. 5.23). This equivalence is used to redefine the $*$-*property* on information flow between confidentiality classes. The new property is denoted the *confidentiality* $*$-*property* and defined as follows.

Definition 5.25 (Confidentiality $*$-property)
Let $b(s:A_1,...,A_k) := \{O : O \in \mathbb{O} \wedge (\forall i \in \{1,...,k\} : (S,O,A_i) \in b)\}$ *where* $k \in \mathbb{N}$, $A_1,...,A_k \in \mathbb{A}$, *and* b *the set of all subjects having access to what objects in what access mode of the respective c-state (cf. Def. B.1).*

A c-state ${}^cV = (b,M,{}^cSCB) \in {}^c\mathbb{V}$ *satisfies the confidentiality* $*$-*property if and only if*

$$\begin{aligned}&\forall S \in \mathbb{S} : b(s:\underline{w},\underline{a}) \neq \emptyset \wedge b(s:r,w) \neq \emptyset\\ &\quad \Rightarrow \forall O_1 \in b(s:r,w), O_2 \in b(s:w,a) : {}^c\overline{scb}(O_1) \mapsto {}^c\overline{scb}(O_2)\,.\end{aligned}$$

with ${}^c\overline{scb}(O_1), {}^c\overline{scb}(O_2) \in {}^cSCB \subseteq {}^c\mathbb{SCB}$ *security binding of object* O_1 *and* O_2, *respectively.*

Analogously to the secure confidentiality condition (cf. Def. 5.24), satisfaction of the confidentiality $*$-*property is defined for c-state sequences, appearances of a c-system, and c-system.*

Remark 5.6 (Partially ordered confidentiality classes) *For plausibility and consistent usage of the* simple-confidentiality property *and* confidentiality $*$-property, *it is important that the confidentiality classes are partially ordered. Otherwise, a comparison of confidentiality classes is not possible and the defined properties are not consistent for* ${}^c\mathbb{SC}$. *Therefore, partially ordered confidentiality classes are a prerequisite for constructing a secure system for controlling information flow with respect to confidentiality.*

This demonstrates that existence of a lattice-based model for information flow control, consolidating the models of Bell and La Padula and Denning. ■

Proof of part (ii): Satisfaction of Denning's Axioms (cf. Theorem 5.2)
Showing Axiom (1): That $({}^c\mathbb{SC},\mapsto)$ is a partial order is implied by the fact $(\mathbb{C},>)$ is totally ordered and $(\mathbb{K},\supset)$ is partially ordered (cf. Def. 5.23).
Showing Axiom (2): ${}^c\mathbb{SC}$ is finite since $\mathbb{C}$ and $\mathbb{K}$ are finite (cf. Def. 5.23).
Showing Axiom (3): With out loss of generality, it is assumed that $(\mathbb{C},>)$ has lower bound $C_L \in \mathbb{C}$ and $(\mathbb{K},\supset)$ has lower bound $K_L \in \mathbb{K}$. Further, let ${}^cSC_L := (C_L,K_L) \in {}^c\mathbb{SC}$ (with out loss of generality). Then, ${}^cSC_L$ is lower bound with ${}^cSC_L \mapsto {}^cSC$ for all ${}^cSC \in {}^c\mathbb{SC}$.

Showing Axiom (4): $\oplus$ is least upper bound operator as shown by Denning [52].

According to Denning [52], the existence of lower bound ${}^{c}SC_L$ and least upper bound operator $\oplus$ implies that greatest lower bound operator $\otimes$ exists, and by satisfaction of all axioms, $({}^{c}\mathbb{SC}, \mapsto, \oplus, \otimes)$ is a lattice. ■

Proof of part (iii): Proving the Security Theorem (cf. Theorem 5.1)
For the combined model, the equivalent of the *Security Theorem* is denoted *Confidentiality Theorem* and is formulated as follows.

Theorem 5.3 (Confidentiality Theorem) *The c-rules ${}^{c}\Omega$ are security preserving and confidentiality $*$-property preserving. Further, a c-system ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ using these c-rules is a secure system and satisfies the confidentiality $*$-property if ${}^{c}z_0$ is a secure c-state which satisfies the confidentiality $*$-property.*

The proof of the *Confidentiality Theorem* consists of two parts:

(a) Showing all c-rules ${}^{c}\rho \in {}^{c}\Omega$ are security preserving and $*$-property preserving:
The rules $\Omega := \{\rho_1, ..., \rho_{10}\}$ are security preserving and $*$-property preserving [127]. Therefore, the c-rules ${}^{c}\rho_i = \rho_i$ for $i \in \{3,5,6,7,9,10\}$ are security preserving and $*$-property preserving, too. According to the definition of confidentiality classes (cf. Def. 5.23), it is easy to see that c-rule ${}^{c}\rho_i$ is equivalent to rule ρ_i for $i \in \{1,2,4\}$ and therefore, security preserving and $*$-property preserving, too. That ${}^{c}\rho_8$ is security preserving and $*$-property preserving is justified by the fact that changing security bindings is equivalent to changing classification and categories. Therefore, ${}^{c}\rho_8$ and ρ_8 are equivalent and both are security preserving and $*$-property preserving. Thus, all c-rules ${}^{c}\rho \in {}^{c}\Omega$ are security preserving and $*$-property preserving.

(b) Showing that ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ is secure and satisfies the confidentiality $*$-property:
Let ${}^{c}Z_0$ initial secure c-state which satisfies the confidentiality $*$-property. This also implies that ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ is secure and satisfies the confidentiality $*$-property. (Proof by induction)

Base step: Let $\|{}^{c}\mathbb{V}\| = 1$ with ${}^{c}z_0$ the only c-state in ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$. Then, all c-states in ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ are secure c-states and satisfy the confidentiality $*$-property.
According to Definition 5.24 (in conj. with Def. 5.18), all c-states in ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ are secure, which implies that all c-state sequences, and all appearances of ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ are secure, and therefore ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ is secure too. Further, according to Definition 5.25 (in conj. with Def. 5.19), all c-states in ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ satisfy the confidentiality $*$-property, which implies that all c-state sequences, all appearances of ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$, and therefore ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ satisfy the confidentiality $*$-property.

Induction step: Let $|{}^{c}\mathbb{V}| = n+1$ with $n \in \mathbb{N}$. Then, there exists c-state ${}^{c}V^* \in {}^{c}\mathbb{V}$ with $|{}^{c}\mathbb{V} \backslash \{{}^{c}V^*\}| = n$ such that all ${}^{c}V \in {}^{c}\mathbb{V} \backslash \{{}^{c}V^*\}$ are secure states and satisfy the confidentiality $*$-property (induction hypothesis). Let ${}^{c}\mathbb{V}^* \subseteq ({}^{c}\mathbb{V} \backslash \{{}^{c}V^*\})$ set of all c-states ${}^{c}V$ where ${}^{c}\rho \in {}^{c}\Omega$ with ${}^{c}\rho({}^{c}R, {}^{c}V) = (D, {}^{c}V^*)$ exists for some ${}^{c}R \in {}^{c}\mathbb{R}$ and $D \in \mathbb{D}$. All rules in ${}^{c}\Omega$ are security preserving and $*$-property preserving and all ${}^{c}V \in {}^{c}\mathbb{V}^* \subseteq ({}^{c}\mathbb{V} \backslash \{{}^{c}V^*\}$ are secure and satisfy the

confidentiality $*$-property. With Definition B.6 it follows that ${}^{c}V^{*}$ is secure and satisfies the confidentiality $*$-property, and therefore, all c-states in ${}^{c}\mathbb{V}$. With Definition 5.24 (in conj. with Def. 5.18) it follows that ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ is secure. Further, according to Definition 5.25 (in conj. with Def. 5.19), ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ satisfies the confidentiality $*$-property. ■

5.2.2.4 Integrity-based information flow control

Integrity-based information flow control was introduced by Biba [22]. In this approach, information flow control is modelled analogously to Bell and La Padula by defining access privileges for reading and writing on objects, classifying subjects and objects, and based on that, defining rules on deciding whether a subject is allowed to have an access privilege. Further, a model for *strict integrity* is constructed in the same way as the model of Bell and La Padula. *Strict integrity* consists of two axioms that are the duals of the *simple-security property* (cf. Def. 5.18) and the $*$-property (cf. Def. 5.19). Therefore, *strict integrity* is fully compatible with the model of [20] and, therefore, information flow control with respect to *strict integrity* and confidentiality can be described in a combined model [175].

In this thesis, information flow control with respect to *strict integrity* is modelled in the same way as the lattice-based model for access control defined in Section 5.2.2.2. In the following the model for integrity-based information flow is constructed.

Model definition (according to Biba [22])
First, a model for integrity levels has to be defined. In the original work by Biba, integrity is modelled in multiple, partially-ordered integrity levels based on the trustworthiness of the information. Without loss of generality, this classification can be replaced by any other *finite* and *partially ordered* set of integrity levels. The *partial order* is required to allow access decisions based on the integrity level as defined by Biba [22], and for consistent construction of a secure system (cf. Remark 5.6). Additionally, *finiteness* is required to satisfy *Denning's Axioms* (cf. Lemma 5.2).

Therefore and without loss of generality, integrity is modelled on the two discrete levels, *low integrity* and *high integrity*, as follows.

Definition 5.26 (Integrity level) *An integrity level* ${}^{i}L \in \{0,1\}$ *is a measure that classifies whether the integrity is low* (${}^{i}L := 0$) *or high* (${}^{i}L := 1$). *Then,* ${}^{i}\mathbb{L} := \{0,1\}$ *is set of integrity levels.*

Based on the integrity level, security classes are defined which are denoted *integrity classes*. Then, analogously to the model for lattice-based information flow control (cf. Lemma 5.1), a model is defined on the basis of integrity classes as follows.

Definition 5.27 (Integrity class) *An integrity class* ${}^{i}SC$ *is a security class corresponding to an integrity level* ${}^{i}L \in {}^{i}\mathbb{L}$, *and* $\overline{int} : {}^{i}\mathbb{SC} \mapsto {}^{i}\mathbb{L}$ *is a function that returns for integrity classes the corresponding integrity level. Then,* ${}^{i}\mathbb{SC} := \{{}^{i}SC_1, ..., {}^{i}SC_n\}$ *is set of integrity classes* ${}^{i}SC_i$ *and* ${}^{i}\mathbb{SC} \subseteq \mathbb{SC}$. *Further, let* (${}^{i}SC_1 \mapsto {}^{i}SC_2$ *for all* ${}^{i}SC_1, {}^{i}SC_2 \in {}^{i}\mathbb{SC}$) *if and only if* ($\overline{int}({}^{i}SC_1) >= \overline{int}({}^{i}SC_2)$). *Additionally,* ${}^{i}\mathbb{SCB}$ *is a set of* integrity security bindings *describing the binding of integrity classes to subjects and objects. Consequently,* i-state ${}^{i}V \in {}^{i}\mathbb{V}$, i-state *sequence*

${}^iZ \in {}^i\mathbb{Z}$, i-request ${}^iR \in {}^i\mathbb{R}$ i-request sequence ${}^iX \in {}^i\mathbb{X}$,*and* i-system ${}^i\Sigma({}^i\mathbb{R}, \mathbb{D}, {}^i\overline{W}, {}^iz_0)$ *are defined analogously to c-state, c-state sequence, c-request, c-request sequence, and c-system by using integrity-based definitions instead of the confidentiality-based ones.*

Further, the *simple-integrity property* and the *integrity $*$-property* are introduced analogously to the simple-confidentiality property (cf. Def. 5.24) and confidentiality $*$-property (cf. Def. 5.25).

Definition 5.28 (Simple-integrity property) *Analogously to simple-confidentiality property (cf. Def. 5.24),* $(S,O,A) \in \mathbb{S} \times \mathbb{O} \times \mathbb{A}$ *satisfies the security integrity condition relative to* ${}^iSCB \subseteq {}^i\mathbb{SCB}$ *if and only if*

(i) $(A = \underline{e}) \vee (A = \underline{a}) \vee (A = \underline{c})$*; and*

(ii) $((A = \underline{r}) \vee (A = \underline{w})) \wedge ({}^i\overline{scb}(O) \mapsto {}^i\overline{scb}(S))$

with ${}^i\overline{scb}(S) \in {}^iSCB$ *security binding of subject S and* ${}^i\overline{scb}(O) \in {}^iSCB$ *security binding of object O.*

Analogously to the simple-security property (cf. Def. 5.18), secure *and* compromise *are defined for i-states, i-state sequences, appearances of an i-system and an i-system.*

Definition 5.29 (Integrity $*$-property)
Let $b(s : A_1,...,A_k) := \{O : O \in \mathbb{O} \wedge (\forall i \in \{1,...,k\} : (S,O,A_i) \in b)\}$ *where* $k \in \mathbb{N}$, $A_1,...,A_k \in \mathbb{A}$*, and b the set of all subjects having access to what objects in what access mode of the respective i-state. Analogously to the confidentiality $*$-property (cf. Def. 5.25), an i-state* ${}^iV = (b, M, {}^iSCB) \in {}^i\mathbb{V}$ *satisfies the integrity $*$-property if and only if*

$$\begin{aligned} &\forall S \in \mathbb{S} : b(s : \underline{w}, \underline{a}) \neq \emptyset \wedge b(s : r, w) \neq \emptyset \\ &\quad \Rightarrow \forall O_1 \in b(s : r, w), O_2 \in b(s : w, a) : {}^i\overline{scb}(O_1) \mapsto {}^i\overline{scb}(O_2) \end{aligned}$$

with ${}^i\overline{scb}(O_1), {}^i\overline{scb}(O_2) \in {}^iSCB \subseteq {}^i\mathbb{SCB}$ *security binding of object* O_1 *and* O_2*, respectively.*

Analogously to the secure integrity condition (cf. Def. 5.28), the satisfaction of the integrity $*$-property is defined for i-state sequences, appearances of an i-system, and i-system.

Based on these properties, rules for a secure system are defined as follows.

Definition 5.30 (10 i-rules for a secure i-system) *An i-rule is defined (analogously to a c-rule; cf. Def. B.6) as a function* ${}^i\rho : \mathbb{R} \times {}^i\mathbb{V} \to \mathbb{D} \times {}^i\mathbb{V}$*. An i-rule maps a request and an i-state to a decision and an i-state. Then, analogous to the definition of* ${}^c\Omega$ *(cf. Def. B.7),* ${}^i\Omega := \{{}^i\Omega_1,...,{}^i\Omega_{10}\}$ *is the set of i-rules for a secure i-system where* ${}^i\rho_i := \rho_i$ *for* $i \in \{3,5,6,7,9,10\}$ *and* ${}^i\rho_1, {}^i\rho_2, {}^i\rho_4, {}^i\rho_8$ *are defined with* ${}^i\overline{scb}(S)$, ${}^i\overline{scb}(O)$, ${}^i\overline{scb}(O') \in {}^i\mathbb{SCB}$ *corresponding security bindings of subject* $S \in \mathbb{S}$ *and objects* $O, O' \in \mathbb{O}$*:*

- ***I-Rule 1 (get-read)*** ${}^i\rho_1$***:*** *A subject S gets read access to an object O if*

  (i) (security preserving) the access attribute $\underline{r}$ *is an element of the corresponding entry of the access matrix, and* ${}^i\overline{scb}(O) \mapsto {}^i\overline{scb}(S)$*; and*

(ii) *(∗-property preserving) for all objects O' where S can write to (i.e., access in append and write mode) is true:* ${}^{i}\overline{scb}(O') \mapsto {}^{i}\overline{scb}(O)$.

- ***I-Rule 2 (get-append)*** ${}^{i}\rho_2$***:*** *A subject S gets append access to an object O if*

  (i) *(security preserving) the access attribute $\underline{a}$ is an element of the corresponding entry of the access matrix; and*

  (ii) *(∗-property preserving) for all objects O' where S can read from to (i.e., access in read and write mode) is true:* ${}^{i}\overline{scb}(O) \mapsto {}^{i}\overline{scb}(O')$.

- ***I-Rule 4 (get-write)*** ${}^{i}\rho_4$***:*** *A subject S gets execute access to an object O if*

  (i) *(security preserving) the access attribute $\underline{w}$ is an element of the corresponding entry of the access matrix, and* ${}^{i}\overline{scb}(O) \mapsto {}^{i}\overline{scb}(S)$*; and*

  (ii) *(∗-property preserving [append]) for all objects O' where S has append access is true:* ${}^{i}\overline{scb}(O') \mapsto {}^{i}\overline{scb}(O)$.

  (iii) *(∗-property preserving [read]) for all objects O' where S has read access is true:* ${}^{i}\overline{scb}(O') \mapsto {}^{i}\overline{scb}(O')$.

  (iv) *(∗-property preserving [write]) for all objects O' where S has write access is true:* ${}^{i}\overline{scb}(O') = {}^{i}\overline{scb}(O)$.

- ***I-Rule 8 (change-*** ${}^{i}SCB$***)*** ${}^{i}\rho_8$***:*** *A subject S can change the security bindings ${}^{i}SCB \subseteq {}^{i}\mathbb{SCB}$ if*

  (i) *(security preserving/∗-property preserving) S changes only security bindings of objects that no subject has access to.*

- ***I-Rules 3, 5, 6, 7, 9, 10*** *are constructed analogously to the rules 3, 5, 6, 7, 9, 10 (respectively) of Ω, since these rules describe general system behaviour, which do not change when modelling integrity*

Having defined the model, it is easy to see that the model satisfies Denning's Axioms (cf. Theorem 5.2) and formulates the Security Theorem (cf. Theorem 5.1) with respect to integrity.

For later use, the lemma on forming a *lattice of integrity classes* and the *Integrity Theorem* (which is the analogon of the *Confidentiality Theorem*; cf. Theorem 5.3) are formulated as follows.

Lemma 5.2 (Lattice of integrity classes) *Given $\mapsto, \oplus, \otimes$ operands on security classes as defined by Denning [52] (cf. Def. 5.22 and Theorem 5.2). Then, $({}^{i}\mathbb{SC}, \mapsto, \oplus, \otimes)$ is a lattice.*

Proof The proof is analogous to part (ii) of the proof of Lemma 5.1 by using integrity definitions instead of confidentiality definitions, and particularly by using the fact that integrity classes are finite and partially ordered. ■

Theorem 5.4 (Integrity Theorem) *Each i-rule in ${}^{i}\Omega$ is security preserving and ∗-property preserving. Further, an i-system ${}^{i}\Sigma({}^{i}\mathbb{R}, \mathbb{D}, {}^{i}\overline{W}, {}^{i}z_0)$ using ${}^{i}\Omega$ is secure and satisfies the integrity ∗-property if ${}^{i}z_0$ is a secure i-state which satisfies the integrity ∗-property.*

Proof The proof is analogous to part (iii) of the proof of Lemma 5.1 by using integrity definitions instead of confidentiality definitions, and particularly by using the fact that $({}^{i}\mathbb{SC}, \mapsto)$ is partially ordered – which is a necessary prerequisite (cf. Remark 5.6) – in conjunction with the *simple-integrity property* (cf. Def. 5.28) and the *integrity* $*$*-property* (cf. Def. 5.29). ■

Remark 5.7 (Generalising the model) *An interesting observation is that the models for integrity and confidentiality are analogous and differ only with respect to the defined security classes (i.e., confidentiality classes and integrity classes) and the allowed information flow between security classes. This is due to the fact that the respective security properties (i.e., simple-confidentiality property, simple-integrity property, and respective* $*$*-properties) and rules (i.e., c-rules and i-rules) are defined based upon allowed information flow instead of comparing security classes of subjects and objects directly. Consequently, modelling access control based on allowed information flow allows the definition of a generalised model that applies to arbitrary types of lattice-based security class where allowed information flow is defined specifically for each type of security class.*

In Section 5.3, such a generalised model is presented which is used in this thesis to model the previously (in Section 5.2.2) introduced properties of confidentiality and integrity as well as the newly (in Section 5.3) introduced properties of location and availability.

5.2.3 Issues on tackling location inhomogeneity in clouds

To address the challenge of location inhomogeneity (cf. Def. 4.6), it is necessary to control the information flow of virtual resources (cf. Remark 5.3) with respect to confidentiality, integrity, availability, and location determination (cf. *Objective 5*; Section 1.3). Without loss of generality, the information flow of virtual resources can be modelled as the access of hardware resources to virtual resources (cf. Remark 5.2). Therefore, hardware resources can be considered subjects (cf. Def. 5.3) and virtual resources can be considered objects (cf. Def. 5.4).

Virtual resources are accessed by hardware resources in several possible ways, which can result in information flow. There exist two types of operation that result in information flow. First, a hardware resource can view a virtual resource, e.g., when loading the virtual machine image or introspecting the virtual machine memory. In this case, information flows from the virtual resource to the hardware resource which is equivalent to access in *read* mode $\underline{r}$ (cf. Def. 5.8). Second, a hardware resource can modify a virtual resource, e.g., when instantiating/deleting a virtual resource or changing its configuration. Then, information flows from the hardware resource to the virtual resource which is equivalent to access in *append* mode $\underline{a}$ (cf. Def. 5.8). It is also possible that hardware resources view and modify virtual resources at the same time (for example, when migrating the virtual resource from one hardware resource to another), which corresponds to accessing in *write* mode $\underline{w}$ (cf. Def. 5.8). Further, there are operations that do not cause information flow. There is no information flow when virtual resources are executed by hardware resources. Possible operations of hardware resources are starting, stopping, suspending, and continuing the execution of virtual resources. These operations correspond to accessing in *execute* mode $\underline{e}$ (cf. Def. 5.8). Additionally, there is no information flow when controlling access to virtual resources. On the one hand, hardware resources can gain or release access to virtual resources, and on the other hand, they can give or

rescind access to virtual resources with respect to other hardware resources. These operations correspond to accessing in *control* mode $\underline{c}$ (cf. Def. 5.8).

Based on these atomic operations, it is possible to describe the more complex operations of the virtual resource management in clouds as follows.

- *Virtual resource creation*: the virtual resource is instantiated by the hardware resource (no access privilege required), and the virtual resource is started by the hardware resource (*execute*).
- *Virtual resource destruction*: the virtual resource is stopped by the hardware resource (*execute*), and the virtual resource is deleted by the hardware resource (*control*).
- *Virtual resource duplication*: the virtual resource is instantiated by the hardware resource by copying a virtual resource from another hardware resource (*write*), and the new virtual resource is started by the copying hardware resource (*execute*).
- *Virtual resource migration*: the virtual resource is created by the hardware resource by copying a virtual resource from another hardware resource (*write*), the new virtual resource is started by the copying hardware resource (*execute*), the copied virtual resource is stopped by the other hardware resource (*execute*), and the copied virtual resource is destroyed by the other hardware resource (*control*).

Consequently, all access operations of hardware resources on virtual resources can be modelled with access modes (cf. Def. 5.8). Therefore, the information flow of virtual resources in clouds can be modelled with respect to the access operations of hardware resources on virtual resources by using the approach of Bell and La Padula. Further, this implies that the models for confidentially-based and integrity-based information flow control presented in Section 5.2.2.2 and Section 5.2.2.4 are applicable, too. In particular, these models apply such that only subjects (i.e., hardware resources and, therefore, hardware providers) gain access to objects (i.e., virtual resources) which comply with the subjects' security classes (i.e., confidentiality classes and integrity classes, respectively). This is due to the fact that information flow is modelled between security classes generally and, therefore, independently of considered objects and subjects. This implies that the models for lattice-based access control presented in Section 5.2.2 are applicable to model information flow control on virtual resources with respect to confidentiality and integrity.

Even though the presented models are applicable, they do not address availability and location determination. Nonetheless, both are important security properties for tackling the challenge of the location inhomogeneity in clouds (cf. Def. 4.6). Hence, the question arises as to how these properties can be modelled in the context of information flow control in general and with respect to virtual resources. In particular, what are the corresponding security classes and how do they apply on controlling the access operations? Existing approaches do not provide answers to these questions (cf. Section 2.4).

In Section 5.3, these open questions will be addressed by introducing a general model of information flow control which covers confidentiality, integrity, availability, and location determination. In particular, security classes and allowed information flows for availability and location of virtual resources will be defined.

5.3 Towards a complete model of information flow control

In Section 5.2, models for lattice-base information flow control on virtual resources war identified good candidates for addressing the challenge of location inhomogeneity (cf. Def. 4.6). Existing approaches focus on single security aspects rather than addressing multiple aspects generally. The first models started in the 1970's with confidentiality [20] [52] and integrity [22]. Since then little progress has been made on introducing new security properties. There are some approaches looking into some aspects of location constraints [201] [166] but, for example, availability as a major security property has remained unaddressed until now. TO address the challenge of location inhomogeneity (cf. Def. 4.6, it is necessary to address information flow control with respect to confidentiality, integrity, availability, and location determination at the same time (cf. *Objective 5*; Section 1.3). Thus, it is necessary to define a general model that is able to address multiple security constraints and allow the introduction of availability and location determination.

In this Section, a general model for lattice-based information flow control is derived from the existing methods. Based on this model, information flow control on virtual resources is modelled with respect to confidentiality, integrity, availability, and location determination. To this end, new methods for modelling availability and location determination are developed. Finally, the application of the general model to the cloud management process is described.

5.3.1 General model on information flow control

Existing models on lattice-based information flow have in common that they can be described analogously with respect to information flow between security classes (cf. Remark 5.7). This observation is used to develop a general model for lattice-based information flow control in this section, which applies to partially ordered security classes generally. The model's construction is based on the models for lattice-based information flow with respect to confidentiality and integrity, which are described in Section 5.2.2. On the one hand, the general model is a consolidation and extension of the existing models for access control developed by Bell and La Padula, Denning, and Biba. On the other hand, restriction in the original design were removed and replaced by more flexible elements, allowing the modelling of multiple security properties at the same time and the introduction of novel security properties beyond those already introduced by this thesis.

To define a general model for information flow control, the following definitions according to the model of Bell and La Padula (cf. Section 5.2.2.1) are used.

- **Subject** $S \in \mathbb{S}$ (cf. Def. 5.3).
- **Object** $O \in \mathbb{O}$ (cf. Def. 5.4).
- **Access attributes** $\mathbb{A} = \{\underline{r}, \underline{w}, \underline{e}, \underline{a}, \underline{c}\}$ (cf. Def. 5.8);
- **Access matrix** $M \in \mathbb{M}$ (cf. Def. 5.9).
- **Decision** $D \in \mathbb{D} = \{\underline{yes}, \underline{no}, \underline{error}, \underline{?}\}$ (cf. Def. 5.15);

- **Decision sequence** $Y \in \mathbb{Y}$, timely ordered sequences (cf. Def. 5.15).
- **Request elements** 'get', 'give', 'release', 'rescind', 'change', 'create', and 'delete' (cf. Def. 5.14).

Further, the following definitions according to Denning's model (cf. Def. 5.22) are used.

- **Security class** $SC \in \mathbb{SC}$.
- **Security binding** $SCB \in \mathbb{SCB} : (\mathbb{S} \cup \mathbb{O}) \times \mathbb{SC}$.
- **Class-combining operator** $\oplus : \mathbb{SCB} \times \mathbb{SCB} \to \mathbb{SCB}$.
- **Flow relation** $\mapsto : \mathbb{SC} \times \mathbb{SC}$.

According to *Denning's Axioms* (cf. Theorem 5.2), $(\mathbb{SC}, \mapsto, \oplus, \otimes)$ is a lattice with upper bound SC_H and lower bound SC_L, and where $\otimes : \mathbb{SCB} \times \mathbb{SCB} \to \mathbb{SCB}$ is least upper bound operator.

Analogously to the models for confidentiality (cf. Section 5.2.2.2) and integrity (cf. Section 5.2.2.4), the following generalised definitions are made for state, state sequence, request, request sequence, and system.

Definition 5.31 (γ-state) *A γ-state ${}^{\gamma}V$ is defined (analogously to c-state; cf. Def. B.1) a triple ${}^{\gamma}V := (b, M, SCB)$ with*

- *$b \subseteq \mathbb{S} \times \mathbb{O} \times \mathbb{A}$ set of all subjects $S \in \mathbb{S}$ having access to objects $O \in \mathbb{O}$ in what access mode, which is described by a set of access attributes $A \subseteq \mathbb{A}$;*
- *$M \in \mathbb{M}$ access matrix in the state ${}^{\gamma}V$; and*
- *$SCB \subseteq \mathbb{SCB}$ set of confidentiality security bindings describing the binding of confidentiality classes to subjects and objects.*

Then, ${}^{\gamma}\mathbb{V}$ is the set of γ-states ${}^{\gamma}V_i$.

Definition 5.32 (γ-state sequence) *A γ-state sequence is (analogously to c-state sequence; cf. Def. B.2) an arbitrary number of timely ordered γ-states ${}^{\gamma}V_i \in {}^{\gamma}\mathbb{V}$. Then, ${}^{\gamma}\mathbb{Z} : {}^{\gamma}\mathbb{V}^{\mathbb{N}}$ is the set of request sequences ${}^{\gamma}Z_i$.*

Definition 5.33 (γ-request) *A γ-request is defined (analogously to c-request; cf. Def. B.3) a quadruple $(S_1, S_2, O_s, {}^{\gamma}G) \subset \mathbb{S}^+ \times \mathbb{S}^+ \times \mathbb{O} \times {}^{\gamma}\mathbb{G}$ with ${}^{\gamma}\mathbb{G} := \mathbb{A} \cup \emptyset \cup \mathcal{P}(\mathbb{SCB})$. Then, ${}^{\gamma}\mathbb{R} : \mathbb{S}^+ \times \mathbb{S}^+ \times \mathbb{O} \times {}^{\gamma}\mathbb{G}$ is the set of requests ${}^{\gamma}R_i$.*

Definition 5.34 (γ-request sequence) *A γ-request sequence is (analogously to c-request sequence; cf. Def. B.4) an arbitrary number of timely ordered γ-requests ${}^{\gamma}R_i \in {}^{\gamma}\mathbb{R}$. Then, ${}^{\gamma}\mathbb{X} : {}^{\gamma}\mathbb{R}^{\mathbb{N}}$ is the set of γ-request sequences ${}^{\gamma}X_i$.*

Definition 5.35 (γ-system) *Let* ${}^{\gamma}\overline{W} \subset {}^{\gamma}\mathbb{R} \times \mathbb{D} \times {}^{\gamma}\mathbb{V} \times {}^{\gamma}\mathbb{V}$. *A* γ*-system* ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R}, \mathbb{D}, {}^{\gamma}\overline{W}, {}^{\gamma}z_0) \subset {}^{\gamma}\mathbb{X} \times \mathbb{Y} \times {}^{\gamma}\mathbb{Z}$ *is defined (analogously to c-system; cf. Def. B.5) by* $({}^{\gamma}X, Y, {}^{\gamma}Z) \in {}^{\gamma}\Sigma({}^{\gamma}\mathbb{R}, \mathbb{D}, {}^{\gamma}\overline{W}, {}^{\gamma}z_0)$ *if and only if* $({}^{\gamma}X_t, Y_t, {}^{\gamma}Z_t, {}^{\gamma}Z_{t-1}) \in {}^{\gamma}\overline{W}$ *for each* $t \in \mathbb{N}$ *where* ${}^{\gamma}z_0 := (\emptyset, M, SCB)$ *is initial state with* $M \in \mathbb{M}$ *initial access matrix and* $SCB \subseteq \mathbb{SCB}$ *initial security bindings.* ${}^{\gamma}\overline{W}$ *is considered* γ-state transition relation.

Further, a generalised form of the *simple-security property* and $*$*-property* is defined analogously to the properties of the models for confidentiality (cf. Section 5.2.2.2) and integrity (cf. Section 5.2.2.4).

Definition 5.36 (General simple-security property) $(S, O, A) \in \mathbb{S} \times \mathbb{O} \times \mathbb{A}$ *satisfies the* general security condition *relative to* $SCB \subseteq \mathbb{SCB}$ *if and only if*

(i) $(A = \underline{e}) \vee (A = \underline{a}) \vee (A = \underline{c})$*; and*

(ii) $((A = \underline{r}) \vee (A = \underline{w})) \wedge (\overline{scb}(O) \mapsto \overline{scb}(S))$

with $\overline{scb}(S) \in SCB$ *security binding of subject* S *and* $\overline{scb}(O) \in SCB$ *security binding of object* O.

Analogously to the simple-security property (cf. Def. 5.18), secure *and* compromise *are defined for* γ*-states,* γ*-state sequences, appearances of a* γ*-system and a* γ*-system.*

Definition 5.37 (General $*$-property)
Let $b(s : A_1, ..., A_k) := \{O : O \in \mathbb{O} \wedge (\forall i \in \{1, ..., k\} : (S, O, A_i) \in b)\}$ *where* $k \in \mathbb{N}$, $A_1, ..., A_k \in \mathbb{A}$, *and* b *the set of all subjects having access to what objects in what access mode of the respective c-state (cf. Def. B.1).*

Analogously to Def. 5.25, a γ*-state* ${}^{\gamma}V = (b, M, SCB) \in {}^{c}\mathbb{V}$ *satisfies the* general $*$-property *if and only if*

$$\forall S \in \mathbb{S} : b(s : \underline{w}, \underline{a}) \neq \emptyset \wedge b(s : r, w) \neq \emptyset \\ \Rightarrow \forall O_1 \in b(s : r, w), O_2 \in b(s : w, a) : \overline{scb}(O_1) \mapsto \overline{scb}(O_2)\,.$$

with $\overline{scb}(O_1), \overline{scb}(O_2) \in SCB \subseteq \mathbb{SCB}$ *security binding of object* O_1 *and* O_2*, respectively.*

Analogously to the general secure condition *(cf. Def. 5.36), the satisfaction of the* general $*$-property *is defined for* γ*-state sequences, appearances of a* γ*-system, and* γ*-system.*

Further, the rules for a secure system are defined analogously to the models for confidentiality (cf. Section 5.2.2.2) and integrity (cf. Section 5.2.2.4).

Definition 5.38 (γ-rule) *A* γ*-rule is (analogously to a c-rule; cf. Def. B.6) a function* ${}^{\gamma}\rho : {}^{\gamma}\mathbb{R} \times {}^{\gamma}\mathbb{V} \to \mathbb{D} \times {}^{\gamma}\mathbb{V}$. *A* γ*-rule maps a* γ*-request and a* γ*-state to a decision and a* γ*-state.*

A γ*-rule* ${}^{\gamma}\rho$ *is* security preserving *if and only if*

$$\forall ({}^{\gamma}R, {}^{\gamma}V) \in {}^{\gamma}\mathbb{R} \times {}^{\gamma}\mathbb{V}\ \exists D \in \mathbb{D}\ \exists\, {}^{\gamma}V' \in {}^{\gamma}\mathbb{V} : \\ {}^{\gamma}\rho({}^{\gamma}R, {}^{\gamma}V) = (D, {}^{\gamma}V') \wedge {}^{\gamma}V \text{ is secure } \gamma\text{-state} \Rightarrow {}^{\gamma}V' \text{ is secure } \gamma\text{-state.}$$

Analogously, a γ-rule ${}^{\gamma}\rho$ is $*$-property preserving *if and only if the γ-state ${}^{\gamma}V$ satisfies the generalised $*$-property implies that γ-state ${}^{\gamma}V'$ satisfies the generalised $*$-property.*

The handling of γ-requests by a γ-rule and the response of a γ-system are defined analogously to rules and systems (cf. Def. 5.20).

Definition 5.39 (10 γ-rules for a secure γ-system) *Analogously to ${}^{c}\Omega$ and ${}^{i}\Omega$ (and therefore, analogously to the rules for a secure system Ω defined by LaPadula et al. [127]), ${}^{\gamma}\Omega := \{{}^{\gamma}\rho_1, ..., {}^{\gamma}\rho_{10}\}$ is the set of γ-rules for a secure γ-system where ${}^{\gamma}\rho_i := \rho_i$ for $i \in \{3,5,6,7,9,10\}$ and ${}^{\gamma}\rho_1, {}^{\gamma}\rho_2, {}^{\gamma}\rho_4, {}^{\gamma}\rho_8$ are defined with $\overline{scb}(S), \overline{scb}(O), \overline{scb}(O') \in \mathbb{SCB}$ respective security bindings of subject $S \in \mathbb{S}$ and objects $O, O' \in \mathbb{O}$:*

- ***γ-Rule 1 (get-read)** ${}^{\gamma}\rho_1$**:** A subject S gets read access to an object O if:*
  - *(i) (security preserving) the access attribute $\underline{r}$ is an element of the corresponding entry of the access matrix, and $\overline{scb}(O) \mapsto \overline{scb}(S)$; and*
  - *(ii) ($*$-property preserving) for all objects O' where S can write to (i.e., access in append and write mode) is true: $\overline{scb}(O) \mapsto \overline{scb}(O')$.*
- ***γ-Rule 2 (get-append)** ${}^{\gamma}\rho_2$**:** A subject S gets append access to an object O if*
  - *(i) (security preserving) the access attribute $\underline{a}$ is an element of the corresponding entry of the access matrix; and*
  - *(ii) ($*$-property preserving) for all objects O' where S can read from (i.e., access in read and write mode) is true: $\overline{scb}(O') \mapsto \overline{scb}(O)$.*
- ***γ-Rule 4 (get-write)** ${}^{\gamma}\rho_4$**:** A subject S gets execute access to an object O if*
  - *(i) (security preserving) the access attribute $\underline{w}$ is an element of the corresponding entry of the access matrix, and $\overline{scb}(O) \mapsto \overline{scb}(S)$;*
  - *(ii) ($*$-property preserving [append]) for all objects O' where S has append access is true: $\overline{scb}(O) \mapsto \overline{scb}(O')$;*
  - *(iii) ($*$-property preserving [read]) for all objects O' where S has read access is true: $\overline{scb}(O') \mapsto \overline{scb}(O)$; and*
  - *(iv) ($*$-property preserving [write]) for all objects O' where S has write access is true: $\overline{scb}(O') = \overline{scb}(O)$.*
- ***γ-Rule 8 (change-SCB)** ${}^{\gamma}\rho_8$**:** A subject S can change the security bindings $SCB \subseteq \mathbb{SCB}$ if*
  - *(i) (security preserving/$*$-property preserving) S changes only security bindings of objects that no subject has access to.*
- ***γ-Rules 3, 5, 6, 7, 9, 10** are constructed analogously to the rules 3, 5, 6, 7, 9, 10 (respectively) of Ω, since these rules describe general system behaviour, which does not change for different security properties.*

Having defined the model, it is possible to formulate the *General Security Theorem* analogously to the *Confidentiality Theorem* (cf. Theorem 5.3) and the *Integrity Theorem* (cf. Theorem 5.4).

Theorem 5.5 (General Security Theorem) *Each γ-rule in ${}^{\gamma}\Omega$ is security preserving and $*$-property preserving. Further, a γ-system ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R},\mathbb{D},{}^{\gamma}\overline{W},{}^{\gamma}z_0)$ using ${}^{\gamma}\Omega$ is secure and satisfies the generalised $*$-property if ${}^{\gamma}z_0$ is a secure γ-state which satisfies the generalised $*$-property.*

Proof The proof is analogous to part (iii) of the proof of Lemma 5.1 by using the generalised definitions instead of confidentiality definitions, and particularly by using the fact that $(\mathbb{SC},\mapsto)$ is partially ordered (which is a necessary prerequisite, cf. Remark 5.6) in conjunction with the *general simple-security property* (cf. Def. 5.36) and the *general $*$-property* (cf. Def. 5.37). ■

Remark 5.8 (Modelling confidentiality and integrity) *Confidentiality can be modelled in the generalised model for information flow control by defining $\mathbb{SC}:={}^{c}\mathbb{SC}$, and this implies that ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R},\mathbb{D},{}^{\gamma}\overline{W},{}^{\gamma}z_0)={}^{c}\Sigma({}^{c}\mathbb{R},\mathbb{D},{}^{c}\overline{W},{}^{c}z_0)$. Integrity can be modelled similarly by defining $\mathbb{SC}:={}^{i}\mathbb{SC}$, which implies that ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R},\mathbb{D},{}^{\gamma}\overline{W},{}^{\gamma}z_0)={}^{i}\Sigma({}^{i}\mathbb{R},\mathbb{D},{}^{i}\overline{W},{}^{i}z_0)$. Further, confidentiality and integrity can be modelled at the same time by using combined security classes that are defined as pairs of confidentiality and integrity classes [175] which are defined by: ${}^{c,i}\mathbb{SC}\subset{}^{c}\mathbb{SC}\times{}^{i}\mathbb{SC}$ with ${}^{c,i}SC_i=({}^{c}SC_i,{}^{i}SC_i)\in{}^{c,i}\mathbb{SC}$ where ${}^{c}SC_i\in{}^{c}\mathbb{SC}$ and ${}^{i}SC_i\in{}^{i}\mathbb{SC}$. Allowed information flow is defined: ${}^{c,i}SC_1\mapsto{}^{c,i}SC_2$ **if and only if** ${}^{c}SC_1\mapsto{}^{c}SC_2$ and ${}^{i}SC_1\mapsto{}^{i}SC_2$ where ${}^{c,i}SC_1,{}^{c,i}SC_2\in{}^{c,i}\mathbb{SC}$, ${}^{c}SC_1,{}^{c}SC_2\in{}^{c}\mathbb{SC}$, and ${}^{i}SC_1,{}^{i}SC_2\in{}^{i}\mathbb{SC}$. Then, $({}^{c,i}\mathbb{SC},\mapsto)$ is partially ordered and finite (due to the fact that ${}^{c}\mathbb{SC}$ and ${}^{i}\mathbb{SC}$ are partially ordered and finite). Therefore,* Denning's Axioms *apply (cf. Theorem 5.2) and $({}^{c,i}\mathbb{SC},\mapsto,\oplus,\otimes)$ is a lattice with lower bound ${}^{c,i}SC_L=({}^{c}SC_L,{}^{i}SC_L)$ and upper bound ${}^{c,i}SC_H=({}^{c}SC_H,{}^{i}SC_H)$.*

5.3.2 Introducing location-determination in information flow control

When modelling allowed information flows of virtual resources in the context of the challenge of location inhomogeneity (cf. Def. 4.6), it is necessary to describe the *ensured level of security* at the location of the hardware resources (i.e., subjects) and how they comply with the *necessary level of security* of virtual resources (i.e., objects). In general, the *ensured level of security* depends on applicable legislation of the country a hardware resource is based in (cf. Section 3.6.1). Therefore, location has to be modelled by using two dimensions: (1) the geographical location of subjects and objects and (2) the geographical application of legislation. For example, for a virtual resource that contains tax data of a German corporate customer that is allowed to be hosted only on hardware resources that are located in Europe, it is necessary to model the location of the hardware resources and virtual resources dependent on the country they are based in.

The geographical application of legislation can be classified by *territories* with their own legislation. Territories can be part of other territories (e.g., Germany is part of the EU/EEA). Further, territories usually do not overlap, i.e., they are fully part of another territory or completely different. However, it is possible for territories to overlap due to contractual agreements, for example, the United Kingdom is a member of the EU/EEA and the Commonwealth

of Nations. Therefore, territories can be modelled as the geographical area with their own legislation, and territories are (partially) ordered by the subset relation of their geographical areas. Additionally, an upper and lower bound for territories can be defined. The upper bound is the *global territory* that contains all other territories. Without loss of generality, the lower bound is defined as the *empty territory* that has an empty intersection with all other territories and, by definition, is contained in every territory. The existence of this territory has no impact on other territories but makes it possible to model the special case where information is not allowed to be part of the system. In the context of IT outsourcing to the cloud, the security calls *empty territory* is used when information is not allowed to flow to the cloud and has to remain at the corporate customer.

The geographical *location* of subjects and objects is specified by a geographical and organisational closed space which is associated with a legal or real person. For example, hardware resources are geographically located at a hosting site operated by a hardware provider. Further, each location is based in one or more hierarchically contained territories. For example, a German hosting site that is located in Passau is based in the following four territories: (1) in the administrative district of Passau, (2) in the federal state of Bavaria, (3) in the German state, and (4) in the EU/EEA.

Consequently, *location*, *territory*, and their interdependency are defined in the model as follows.

Definition 5.40 (Location) *A* location *loc is defined as a geographical and organisational closed space where specified subjects and objects are located, and* $\mathbb{LOC} := \{loc_1, ..., loc_n\}$ *set of locations* loc_i.

Definition 5.41 (Territory) *A* territory T *is defined as a geographical area having its own legislation, and* $\mathbb{T} := \{T_1, ..., T_n\}$ *set of territories* T_i. *Further,* $\overline{loc} : \mathbb{T} \rightarrow \mathcal{P}(\mathbb{LOC})$ *is the function returning the set* $LOC_T \subset \mathbb{LOC}$ *where all locations* $loc \in LOC_T$ *are geographically located in territory* T. *In addition,* $T_{global} \in \mathbb{T}$ *with* $\overline{loc}(T_{global}) = \mathbb{LOC}$ *is defined as the global territory containing all locations, and* $T_{local} \in \mathbb{T}$ *with* $\overline{loc}(T_{local}) = \emptyset$ *is defined as the empty territory containing no location. Then,* $\mathbb{T}$ *is partially ordered by the* $\subset$*-relation on* $\overline{loc}$ *with upper bound* T_{global} *and lower bound* T_{local}. *Further,* $\mathbb{T}$ *is finite when modelling real-world systems.*

Based on the definition of *location* and *territory*, it is possible to define location classes and the information flow allowed between them. For each territory, a location class is defined. Then, a location class corresponds to the *ensured level of security* within a specific territory. Further, the information flow allowed between location classes depends on the compatibility of their *ensured level of security*. In general, the *ensured level of security* of a territory is compatible with that of another territory only if in the first territory the legislation of the second applies. This is the case only if the first territory is contained in the second territory. For example, the *ensured level of security* in Germany is compatible with that in the EU/EEA since Germany is a member state of the EU/EEA, and basically the legislation of the EU/EEA also applies to Germany. On the other hand, the *ensured level of security* in the EU/EEA does not necessarily have to be compatible with that in Germany since Germany is not the only member state and German legislation is not applicable in other member states generally. This

implies that information flow between location classes is allowed only if the territory of the target location class lies in the territory of the source location class. For example, information is allowed to flow from Germany to France if the information is classified to flow within the EU/EEA. Consequently, location classes and the information flow allowed between them are defined as follows.

Definition 5.42 (Location class) *A* location class ${}^{loc}SC_T$ *is defined as the corresponding security class to territory* $T \in \mathbb{T}$*, and* ${}^{loc}\mathbb{SC} \subseteq \mathbb{SC}$ *is defined as a set of location classes* ${}^{loc}SC_T$*.*

Also, let ${}^{loc}SC_{global} \in {}^{loc}\mathbb{SC}$ *the corresponding security class to the* global territory T_{global}*,* ${}^{loc}SC_{local} \in {}^{loc}\mathbb{SC}$ *the corresponding security class to the* empty territory T_{local}*.*

In addition, let $({}^{loc}SC_{T_1} \mapsto {}^{loc}SC_{T_2})$ ***if and only if*** $(\overline{loc}(T_1) \supseteq \overline{loc}(T_2))$ *where* $T_1, T_2 \in \mathbb{T}$ *and* ${}^{loc}SC_{T_1}, {}^{loc}SC_{T_2} \in {}^{loc}\mathbb{SC}$ *(i.e., information in* ${}^{loc}SC_{T_1}$ *is allowed to flow to* ${}^{loc}SC_{T_2}$ *if and only if all locations in territory* T_2 *also lie in territory* T_1*).*

For an object or subject, the location class can be interpreted as a boundary of their possible locations. Thereby, it is possible to model the geographical location of subjects and objects.

Further, because territories are finite and partially ordered, it follows that $({}^{loc}\mathbb{SC}, \mapsto)$ is also finite and partially ordered, and ${}^{loc}SC_{global}$ is the corresponding lower bound and ${}^{loc}SC_{local}$ is the upper bound. This implies that *Denning's Axioms* apply to location classes and $({}^{loc}\mathbb{SC}, \mapsto, \oplus, \otimes)$ is a lattice.

In addition, $({}^{loc}\mathbb{SC}, \mapsto)$ is partially ordered and implies that ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R}, \mathbb{D}, {}^{\gamma}\overline{W}, {}^{\gamma}z_0)$ with $\mathbb{SC} = {}^{loc}\mathbb{SC}$ is a secure system satisfying the general $*$-property (cf. Theorem 5.5). This is made plausible by looking at the *general simple-security property* (cf. Def. 5.36) and the *general* $*$*-property* (cf. Def. 5.37). The *general simple-security property* defines allowed information flow from objects to subjects. Information can flow from object $O \in \mathbb{O}$ to subject $S \in \mathbb{S}$ only for the access modes read and write (since viewing the object causes information flow to the subject). Let ${}^{loc}SC_{T_O} \in {}^{loc}\mathbb{SC}$ be the security class of O and ${}^{loc}SC_{T_S} \in {}^{loc}\mathbb{SC}$ the security class of S with $T_O, T_S \in \mathbb{T}$ being the corresponding territories. Then, according to the definition of location classes, information flow $({}^{loc}SC_{T_O} \mapsto {}^{loc}SC_{T_S})$ is allowed if and only if $(\overline{loc}(T_O) \supseteq \overline{loc}(T_S))$. This means that information flow is allowed if and only if the possible locations of the object covers the possible locations of the subject. This implies that objects are accessible only by subjects that are based at locations that are also allowed for the object. The object's *necessary level of security* is therefore satisfied by the subject's *ensured level of security*. The *general* $*$*-property* defines the information flow allowed between objects via subjects and, therefore, the information flow from subjects to objects. Here, the access of a subject to objects is allowed if and only if information is allowed to flow from objects which are accessed in read and write mode to objects that are accessed in write and append mode. With respect to location classes, information flow is allowed if and only if the possible locations of the viewed object covers the possible locations of the modified objects. This implies that the *necessary level of security* of viewed objects is satisfied by the *necessary level of security* of modified objects. Consequently, the location classes ${}^{loc}\mathbb{SC}$ satisfy the *General Security Theorem* (cf. Theorem 5.5).

Remark 5.9 (Modelling confidentiality, integrity, and location) *Analogously to Remark 5.8, confidentiality, integrity, and location can be modelled at the same time by using combined security classes that are defined as triples of confidentiality, integrity, and location classes which*

are defined by: ${}^{c,i,loc}\mathbb{SC} \subset {}^{c}\mathbb{SC} \times {}^{i}\mathbb{SC} \times {}^{loc}\mathbb{SC}$ *with* ${}^{c,i,loc}SC_i = ({}^{c}SC_i, {}^{i}SC_i, {}^{loc}SC_i) \in {}^{c,i,loc}\mathbb{SC}$ *where* ${}^{c}SC_i \in {}^{c}\mathbb{SC}$, ${}^{i}SC_i \in {}^{i}\mathbb{SC}$, *and* ${}^{loc}SC_i \in {}^{loc}\mathbb{SC}$.

Allowed information flow is defined: ${}^{c,i,loc}SC_1 \mapsto {}^{c,i,loc}SC_2$ *if and only if* ${}^{c}SC_1 \mapsto {}^{c}SC_2$ *and* ${}^{i}SC_1 \mapsto {}^{i}SC_2$ *and* ${}^{loc}SC_1 \mapsto {}^{loc}SC_2$ *where* ${}^{c,i,loc}SC_1, {}^{c,i,loc}SC_2 \in {}^{c,i,loc}\mathbb{SC}$, ${}^{c}SC_1, {}^{c}SC_2 \in {}^{c}\mathbb{SC}$, ${}^{i}SC_1, {}^{i}SC_2 \in {}^{i}\mathbb{SC}$, *and* ${}^{loc}SC_1, {}^{loc}SC_2 \in {}^{loc}\mathbb{SC}$.

Then, $({}^{c,i,loc}\mathbb{SC}, \mapsto)$ *is partially ordered and finite (due to the fact that* ${}^{c}\mathbb{SC}$, ${}^{i}\mathbb{SC}$, *and* ${}^{loc}\mathbb{SC}$ *are partially ordered and finite). Therefore,* Denning's Axioms *apply (cf. Theorem 5.2) and* $({}^{c,i,loc}\mathbb{SC}, \mapsto, \oplus, \otimes)$ *is a lattice with lower bound* ${}^{c,i,loc}SC_L = ({}^{c}SC_L, {}^{i}SC_L, {}^{loc}SC_L)$ *and upper bound* ${}^{c,i,loc}SC_H = ({}^{c}SC_H, {}^{i}SC_H, {}^{loc}SC_H)$.

5.3.3 Introducing availability in information flow control

Another important security property for addressing the challenge of location inhomogeneity (cf. Def. 4.6) is the availability of virtual resources. When being utilised, virtual resources have to be accessible and functional. Both depend on several factors, which are internal and external to the virtual resources. The accessibility depends, for example, on the connectivity of the access network (external) and the functioning of the connection end-points of the virtual resources (internal). The functioning of virtual resources depends, for example, on the functioning of the hosting hardware resources (external) and the applications running on them (internal). For the information flow of virtual resources, external factors that influence the availability of virtual resources can change, while internal factors do not change due to migration. For example, the migration of a virtual resource from a highly available hardware resource to another with lower availability also reduces the availability of the virtual resource, but the migration itself does not change the state of applications running on the virtual machine (assuming that downtimes during migration are negligible and virtual resources are fully functioning after migration which is possible when using live migration [72]). With respect to information flow of virtual resources, availability can be modelled as a requirement for a subject (i.e., hardware resource) when gaining access to an object (i.e, virtual resource). For that reason, objects are classified by their required availability and subjects are classified by their provided availability.

To model classifications by availability the following definition of availability is used.

Definition 5.43 (Availability [221]) *According to Xie et al. [221, pp. 11–12],* availability $A(t)$ *is defined as the probability that a system is up at a given time t. Further, the* asymptotic availability *is given by*

$$A = \lim_{t \to \infty} A(t) = \frac{\textit{system up time}}{\textit{system up time} + \textit{system down time}} = \frac{MTTF}{MTTF + MTTR}$$

where MTTF is the mean time to failure *of a system, and MTTR is the* mean time to repair *of a system. In the following, the term* availability *is used in the sense of* asymptotic availability.

Remark 5.10 (Availability is totally ordered and finite) *According to the Def. 5.43, availability is represented by a value of the interval* $[0,1]$*, and therefore, it is totally ordered by the* $<$*-relation. Theoretically, there are infinite values for availability, but in practice, only a*

finite number of availabilities are relevant. The reason is that availability is used in practice by the following scheme: 90%, 95%, 99%, 99.5%, 99.9%, 99.95%, 99.99%, 99.995%, ..., 100%. Further, the highest availabilities that are used can be found in function safety standards and are not higher then 0.99999%, which is the upper boundary of safety integrity level 4 (based on IEC 61508 [108]). Therefore, the number of relevant availabilities in practice is finite. This makes availability (according to Def. 5.43) a good candidate for classifying subjects and objects in a lattice-based model for information flow control.

In the model, the required availability of an object is classified by the minimum availability that has to be provided by an accessing subject. Analogously, the provided availability of a subject is classified by the minimum availability that is provided by the subject. Consequently, *availability classes* are defined as follows:

Definition 5.44 (Availability class) *An* availability class ${}^{av}SC_x := [x,1]$ *with* $x \in [0,1]$ *is defined as a continuous interval of (asymptotic) availabilities* $A \in {}^{av}SC_x$*. Then,* ${}^{av}\mathbb{SC}$ *is set of availability classes* ${}^{av}SC_x$*. Further, let* ${}^{av}SC_{x_1} \mapsto {}^{av}SC_{x_2}$ *for all* $x_1, x_2 \in [0,1]$ *and* ${}^{av}SC_{x_1}, {}^{av}SC_{x_2} \in {}^{av}\mathbb{SC}$ *if and only if* ${}^{av}SC_{x_1} \supseteq {}^{av}SC_{x_2}$*, i.e., information in* ${}^{av}SC_{x_1}$ *is allowed to flow to* ${}^{av}SC_{x_2}$ *if and only if availability* x_1 *is less or equal availability* x_2*.*

Then, $({}^{av}\mathbb{SC}, \mapsto)$ is partially ordered since $\subset$-relation partial order and availabilities are totally ordered by the $<$-relation. Further, ${}^{av}\mathbb{SC}$ is finite for availabilities that are relevant in practice (cf. Remark 5.10). In addition, ${}^{av}SC_{100\%} \in {}^{av}\mathbb{SC}$ is upper bound and ${}^{av}SC_{0\%} \in {}^{av}\mathbb{SC}$ is lower bound. This implies that *Denning's Axioms* apply for availability classes and $({}^{av}\mathbb{SC}, \mapsto, \oplus, \otimes)$ is a lattice.

Moreover, the fact that $({}^{av}\mathbb{SC}, \mapsto)$ is partially ordered implies that ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R}, \mathbb{D}, {}^{\gamma}\overline{W}, {}^{\gamma}z_0)$ with $\mathbb{SC} = {}^{av}\mathbb{SC}$ is a secure system satisfying the general $*$-property (cf. Theorem 5.5). Analogously to location classes (cf. Section 5.3.2), this is made plausible by looking at the *general simple-security property* (cf. Def. 5.36) and the *general* $*$*-property* (cf. Def. 5.37). The general simple-security property defines allowed information flow from objects to subjects, which is caused by viewing objects. This is allowed only if the object's availability class contains the subject's availability class, i.e., the availability provided by the subject is higher then the required availability of the object. Thereby, the subject can ensure the required availability of the object – which is the intention of information flow control with respect to availability. The *general* $*$*-property* defines the information flow allowed between objects via subjects (and, therefore, from subjects to objects), which is caused by modifying objects. Here, the access of a subject to an object is allowed only if the availability of objects that are modified is higher than the availability of objects that are viewed. This implies that information can flow only from lower availability classes to higher availability classes – which is again the intention of information flow control with respect to availability. Consequently, the availability classes ${}^{av}\mathbb{SC}$ satisfy the *General Security Theorem* (cf. Theorem. 5.5).

Remark 5.11 (Deletion and availability) *When deleting an object, the availability of the object is no longer given. Therefore, deletion is allowed only if the availability of the object is no longer needed. The information model addresses this in the rule* ${}^{\gamma}\rho_{10}$ *for deleting objects by allowing the deletion of objects only where no other subject has access to those objects. As*

long as at least one subject has access to an object, the deletion of the object is forbidden. In such a case, the object is said to be "locked". This mechanism of locking objects can be used to control the deletion of objects. In addition, a new subject having the object's security class is added to the system ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R}, \mathbb{D}, {}^{\gamma}\overline{W}, {}^{\gamma}z_0)$. *The new subject is used to lock the object (or its backup instance) as long as it has to be available. It is possible then to control the deletion of objects by giving and rescinding access privileges to the locking subject.*

Remark 5.12 (Modelling confidentiality, integrity, availability, and location) *Analogously to Remark 5.8 and Remark 5.9, confidentiality, integrity, availability, and location can be modelled at the same time by using combined security classes that are defined as quadruples of confidentiality, integrity, availability, and location classes which are defined by:* ${}^{c,i,av,loc}\mathbb{SC} \subset {}^{c}\mathbb{SC} \times {}^{i}\mathbb{SC} \times {}^{av}\mathbb{SC} \times {}^{loc}\mathbb{SC}$ *with* ${}^{c,i,av,loc}SC_i = ({}^{c}SC_i, {}^{i}SC_i, {}^{av}SC_i, {}^{loc}SC_i) \in {}^{c,i,av,loc}\mathbb{SC}$ *where* ${}^{c}SC_i \in {}^{c}\mathbb{SC}$, ${}^{i}SC_i \in {}^{i}\mathbb{SC}$, ${}^{av}SC_i \in {}^{av}\mathbb{SC}$, *and* ${}^{loc}SC_i \in {}^{loc}\mathbb{SC}$.

Allowed information flow is defined: ${}^{c,i,a,loc}SC_1 \mapsto {}^{c,i,a,loc}SC_2$ *if and only if* ${}^{c}SC_1 \mapsto {}^{c}SC_2$ *and* ${}^{i}SC_1 \mapsto {}^{i}SC_2$ *and* ${}^{av}SC_1 \mapsto {}^{av}SC_2$ *and* ${}^{loc}SC_1 \mapsto {}^{loc}SC_2$ *where* ${}^{c,i,a,loc}SC_1, {}^{c,i,a,loc}SC_2 \in {}^{c,i,av,loc}\mathbb{SC}$, ${}^{c}SC_1, {}^{c}SC_2 \in {}^{c}\mathbb{SC}$, ${}^{i}SC_1, {}^{i}SC_2 \in {}^{i}\mathbb{SC}$, ${}^{av}SC_1, {}^{av}SC_2 \in {}^{av}\mathbb{SC}$, *and* ${}^{loc}SC_1, {}^{loc}SC_2 \in {}^{loc}\mathbb{SC}$.

Then, $({}^{c,i,av,loc}\mathbb{SC}, \mapsto)$ *is partially ordered and finite (due to the fact that* ${}^{c}\mathbb{SC}$, ${}^{i}\mathbb{SC}$, ${}^{av}\mathbb{SC}$, *and* ${}^{loc}\mathbb{SC}$ *are partial ordered and finite). Therefore,* Denning's Axioms *apply (cf. Theorem 5.2) and* $({}^{c,i,av,loc}\mathbb{SC}, \mapsto, \oplus, \otimes)$ *is a lattice with upper bound* ${}^{c,i,av,loc}SC_L = ({}^{c}SC_L, {}^{i}SC_L, {}^{av}SC_L, {}^{loc}SC_L)$ *and lower bound* ${}^{c,i,av,loc}SC_H = ({}^{c}SC_H, {}^{i}SC_H, {}^{av}SC_H, {}^{loc}SC_H)$.

5.3.4 Information flow control in the cloud management process

In Section 5.3.1 a general model for lattice-based information flow control was introduced which applies to modelling the security properties that are essential for addressing the challenge of location inhomogeneity (cf. Def. 4.6), namely, confidentiality and integrity (cf. Remark 5.8), availability (cf. Section 5.3.3), and location determination (cf. Section 5.3.2). To address the challenge of location inhomogeneity in clouds, the control of the information flow of virtual resources has to be modelled and decisions on allowed information flow have to be integrated into the resource management of cloud infrastructures. The resource management of cloud infrastructures is formally described in the IaaS cloud computing entity-relationship model in Section 4.1. In particular, the *cloud management process* $\overline{cmp} : \mathcal{P}(\mathbb{VR}) \times \mathcal{P}(\mathbb{HW})$ (cf. Section 4.1.4.4) maps virtual resources $\mathbb{VR}$ to hardware resources $\mathbb{HW}$. This makes is possible to describe for each virtual resource $\overline{vr} \in \mathbb{VR}$ the hosting hardware resource $\overline{hr} = \overline{cmp}(\overline{vr}) \in \mathbb{HW}$.

In this section, the application of the lattice-based model for information flow control on the cloud management process is presented. Further, it is shown that if decisions of the resource management are modelled as the cloud management process then the validity of these decisions is enforceable by security classes defined on virtual resources and hardware resources.

5.3.4.1 Application of lattice-based information flow control on virtual resources

According to the considerations on modelling *information flow of virtual resources* in Section 5.2.3, hardware resources are modelled as subjects, i.e., $\mathbb{S} = \mathbb{HW}$, and virtual resources

are modelled as objects, i.e., $\mathbb{O} = \mathbb{VR}$. Further, security bindings for virtual resources are defined according to the corporate customers' requirements on confidentiality, integrity, availability, and location-determination using corresponding security classes $\mathbb{SC} := {}^{c,i,av,loc}\mathbb{SC}$ covering these properties (cf. Remark 5.12). Analogously, security bindings for hardware resources are defined according to their provided security properties. Further, the access operations of hardware resources on virtual resources are described as follows.

- r (read): virtual resource is viewed by hardware resource (e.g., loading virtual machine image).
- a (append): virtual resource is modified by hardware resource (e.g., configuring virtual resource).
- w (write): virtual resource is viewed and modified by hardware resource (e.g., coping virtual resource).
- e (execute): virtual resource is executed by hardware resource (e.g., starting virtual machine instance).
- c (control): hardware resource extends access to other hardware resources (e.g., migration triggered by hardware resource operating as *compute manager*, cf. Section 4.1.4.1).

Access requests by hardware resources are modelled as *request elements* (cf. Def. 5.14) and *requests* (cf. Def. 5.12). For example, when hardware resource $\overline{hw} \in \mathbb{HW}$ has to configure a virtual resource $\overline{vr} \in \mathbb{VR}$, the corresponding request element is *get* with request $(\emptyset, \overline{hw}, \overline{vr}, \underline{a}) \in {}^{\gamma}\mathbb{R}$ (cf. Def 5.33). Then, rule ${}^{\gamma}\rho_2$ applies.[1]

Further, there are operations on virtual resources that are controlled by the cloud management process, namely, creation, destruction, duplication, and migration. According to the considerations in Section 5.2.3, these operations can be described by combined access operations as follows.

- *Virtual resource creation*: hardware resource accesses virtual resource in *execute* mode, in particular covered by the access element *create* (cf. Def. 5.14);
- *Virtual resource destruction*: hardware resource accesses virtual resource in *execute* and *control* mode, in particular covered by the access element *delete* (cf. Def. 5.14);
- *Virtual resource duplication*: hardware resource accesses virtual resource in *write*[2] and *execute* mode;

[1] According Bell and La Padula [127], the first subject of a request has to be empty when requesting access in append mode.

[2] Copying a virtual resource includes viewing the virtual resource at the source hardware resource and modifying it at the target hardware resource. This includes the creation of a new virtual resource at the target hardware resource. The new virtual resource is a copy of the virtual resource at the source hardware resource. A copy is intentionally identical to the original version, but does not necessarily have to be, for example, due to bit errors during the copying process or malicious modification. Therefore, copying virtual resources is modelled as access in *write* mode.

- *Virtual resource migration*: hardware resource accesses virtual resource in *write* and *execute* mode (copying and starting new virtual resource), and hardware resource accesses virtual resource in *execute* and *control* mode (stopping and deleting original virtual resource).

The access requests of the hardware resources necessary for each access operation are again modelled as *request elements* (cf. Def. 5.14) and *requests* (cf. Def. 5.12). Additionally, the cloud management process controlling the combined access operations has to be modelled, which is described in Section 5.3.4.2.

Then, the system ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R},\mathbb{D},{}^{\gamma}\overline{W},{}^{\gamma}z_0)$ (cf. Def. 5.35) with the rules ${}^{\gamma}\Omega$ (cf. Def. 5.39) describes the access of hardware resources to virtual resources based on the information flow allowed between security classes of subjects and objects.

5.3.4.2 Modelling information flow control in the cloud management process

In cloud infrastructures, the *information flow of virtual resources* is coordinated by the *cloud management process* $\overline{cmp}$ (cf. Section 4.1.4.4). The cloud management process assigns virtual resources to hardware resources. To decide and enforce the allowed information flow, this assignment has to comply with the decisions of the system ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R},\mathbb{D},{}^{\gamma}\overline{W},{}^{\gamma}z_0)$. To achieve this, the operations on virtual resources controlled by the cloud management process (namely, creation, destruction, duplication, and migration) are modelled as request elements (cf. Def. 5.14) and *requests* (cf. Def. 5.12) in the system ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R},\mathbb{D},{}^{\gamma}\overline{W},{}^{\gamma}z_0)$, and the decisions (cf. Def. 5.15) of ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R},\mathbb{D},{}^{\gamma}\overline{W},{}^{\gamma}z_0)$ are applied to the cloud management process enforcing the allowed information flow. The modelling of each operation on virtual resources controlled by the cloud management process is described in the following.

The **creation of virtual resources** is modelled as the request element *create* (cf. Def. 5.14). For example, if a virtual machine $\overline{vm} \in \mathbb{VM}$ is requested by a corporate customer, the cloud management processes selects a *compute server* with hypervisor $\overline{cs}_{\overline{hs}} \in \mathbb{CS}$ at hosting site $\overline{hs} \in \mathbb{HS}$ for hosting $\overline{vm}$. In ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R},\mathbb{D},{}^{\gamma}\overline{W},{}^{\gamma}z_0)$, $\overline{cs}_{\overline{hs}}$ is subject $S_{\overline{cs}_{\overline{hs}}} \in \mathbb{S}$ and $\overline{vm}$ is object $O_{\overline{vm}} \in \mathbb{O}$. Since a new virtual machine is requested and should be started (i.e., executed) by the selected *compute server*, the corresponding request element is *create* (cf. Def. 5.14) with the corresponding request $(\emptyset, S_{\overline{cs}_{\overline{hs}}}, O_{\overline{vm}}, \underline{e}) \in {}^{\gamma}\mathbb{R}$ (cf. Def 5.33). Then, rule ${}^{\gamma}\rho_9$ applies.[1]

When creating new objects, it is necessary to classify them (i.e., assigning a security class). By default, the security class of the creating subject is assigned to the created object, which is reasonable for ensuring a secure system [127]. In the context of cloud computing, the security class of the object is requested by corporate customers, which do not necessarily have to comply with the security class of assigned hardware resources. Therefore, it is necessary to ensure that information flow from the requested security class to the security class of the hardware resource is allowed (security preserving; cf. Def. 5.36), and that information flow from the requested security class to the security class of other virtual resources that are modified by the hardware resource is allowed ($*$-property preserving; cf. Def. 5.36). A possible solution is to select only hardware resources with security classes where information flow is security

[1] According Bell and La Padula [127], the first subject of a request has to be empty when requesting the object creation.

preserving and $*$-property preserving. Then, the virtual resource is created by the hardware resource (i.e., *create*) and, afterwards, the security class is set to the security class requested by the corporate customer (i.e., *change*). It is also possible to combine the two requests to a single request by introducing a new request element $create^+$ for creating objects with assigned security classes. The corresponding rule is derived from the existing rule ${}^\gamma\rho_9$ as follows.

- **Rule 9^+ ($create^+$-object-with-assigned-security-class) ρ_9^+:** A subject S can create an object O where $\overline{scb}(O) = sc$ with $\overline{scb}(O) \in \mathbb{SCB}$ security binding of object O and $sc \in \mathbb{SC}$ assigned security class if

  (i) (security preserving/$*$-property preserving) O does not exist;

  (ii) (security preserving) $\overline{scb}(O) \mapsto \overline{scb}(S)$; and

  (iii) ($*$-property preserving) for all objects O' where S can write to (i.e., access in append and write mode) is true: $\overline{scb}(O) \mapsto \overline{scb}(O')$.

By design, the rule ρ_9^+ is security preserving (cf. Def. 5.36) and $*$-property preserving (cf. Def. 5.37). The extended set of rules including ρ_9^+ is denoted: ${}^\gamma\Omega^+ := {}^\gamma\Omega \cup \{\rho_9^+\}$. Then, the system ${}^\gamma\Sigma({}^\gamma\mathbb{R}, \mathbb{D}, {}^\gamma\overline{W}, {}^\gamma z_0)$ with the rules ${}^\gamma\Omega^+$ is secure and satisfies the $*$-property.

The **destruction of a virtual resource** is modelled as the request element *delete*. There is no information flow during the deletion of a virtual resources. Therefore, objects are deleted independently of their security class, and the rule ${}^\gamma\rho_{10}$ applies without modifications. To ensure availability, it might be necessary to prevent deletion by "locking" virtual resources using a hardware resource sustaining access to the virtual resource (cf. Remark 5.11). A possible implementation in the cloud management process is that a specific hardware resource is assigned to operate the virtual resource (or a backup instance) which is deleted only on the request of the corporate customer (or when the availability of the virtual resource is no longer needed).

Remark 5.13 (Deletion through hardware failure) *It is possible that the failure of a hardware resource results in the deletion of virtual resources that are running upon them. For example, the hardware resource is physically destroyed. The resulting deletion of the virtual resource is an illegal operation in the information model since the deletion of objects is caused by the removal of subjects and not by a requested access of a subject. Such dependency of subjects and objects is specific to the information flow of virtual resources and is not covered by the general model on information flow. Therefore, the deletion of virtual resources due to the failure of hardware resources is a covert channel in the model undermining the availability property. This covert channel can be addressed in the cloud management process by assigning recovery functions to hardware resources (cf. fault management in Section 4.1.3.2) which ensure the recovery of virtual resources in the even of such deletions. However, the mitigation/prevention of the covert channel depends on the effectiveness of the recovery functions.*

The **duplication of virtual resources** is modelled as the request element *get* requesting *write* access in combination with the request element $create^+$. This is made plausible by looking at the four steps that are necessary to duplicate a virtual resources. First, the hardware resource requires access to the origin virtual resource in *read* mode (step 1). Second, the hardware resource has to create a new virtual resource with the same security class as that of the

origin virtual resource (step 2). Third, the hardware resource requires access to the created virtual resource in *append* mode for copying data and states of the origin virtual resource (step 3). Finally, after completion of the copy process, the hardware resource requires access to the created virtual resource in *execute* mode for starting it (step 4). Therefore, a subject requires access in *read*, *append*, and *execute* mode to complete the duplication. The duplication is only possible if all necessary access privileges are granted and information flow between the subject's and object's security class is allowed according to the corresponding rules (i.e., ρ_1 and ρ_2). The access in *execute* mode depends solely on the granted *execute* access privilege for the newly created virtual resource, since execute is allowed independently of the subject's and object's security class. Further, the *execute* access privilege can be assigned on object creation by the creating subject. Thus, access in *execute* mode is allowed if *execute* access privilege is assigned on object creation. The access in *read* and *append* mode depends on granted access privileges and allowed information flow between the subject's and object's security class. Both, origin object and duplicate object have the same security class. Therefore, access to the duplicate object in *append* mode is allowed if and only if access to the origin object in append mode is allowed. Further, access in *read* and *append* mode is equivalent to access in *write* mode. This implies that access to the origin object in *write* mode is allowed if and only if access to the origin object in *read* mode and access to the duplicate object in *append* mode is allowed. Consequently, all access operations necessary for the duplication are allowed if access to the origin object in *write* mode is allowed and *execute* access privilege is assigned on object creation. If a subject does not have the *write* access privilege on the origin object or *execute* is assigned when the object is created, the duplication fails and is aborted. On abortion, the system is still secure which is made plausible by looking at each step. If duplication fails at step 1 then the subject is not allowed to access the origin object in read mode (i.e., ${}^{\gamma}\rho_1$ return the decision *no*). If duplication fails at step 2 the subject is allowed to access the origin object in *read* mode but is not allowed to create a new object (i.e., ${}^{\gamma}\rho_9$ return the decision *no*). If duplication fails at step 3 the subject is allowed to access the origin object in *read* mode and to create a new object but is not allowed to access the created object in *append* mode (i.e., ${}^{\gamma}\rho_2$ return the decision *no*). Since the access privilege *append* is set when the object is created, this can happen only if the created object has a security class to which information is not allowed from security classes of other objects that the subject has access to. This implies that a wrong security class was assigned on creation in step 2. However, the system is secure since no information flow is allowed. If duplication fails on step 3 then the subject is allowed to access the origin object in *read* mode, to create the new objects, and to access the new object in *append* mode but is not allowed to access the new object in *execute* mode (i.e., ${}^{\gamma}\rho_3$ return the decision *no*). This only can happen if *execute* access privilege was not set when the object was created in step 2. Since there is no information flow on access in *execute* mode, the system is secure, independently of whether it has *execute* access privilege set. Moreover, it is possible to manually set the *execute* access privilege by the cloud management process, to repeat step 4, and to successfully complete the duplication.

The **migration of virtual resources** is modelled similarly to the duplication of virtual resources by the request element *get* requesting *write* and *execute* access in combination with the request elements $create^+$ and *delete*. This is made plausible by looking at the six steps that

are necessary to migrate a virtual resource. The first four steps are identical to the duplication of virtual resources. After duplication the hardware resources require access to the origin virtual resources in *execute* mode to stop it (step 5). Then, the hardware resource requires access in *control* mode to delete the origin virtual resource (step 6). Therefore, additionally to duplication, the *execute* and *control* access privileges for the origin virtual resource are required and the deletion of the origin virtual resource must be possible. Otherwise, the migration is aborted after duplication. Consequently, before starting duplication, the subject that requests access for migrating the object must have the *write*, *execute*, and *control* access privileges for the origin object. As with the duplication, the system is secure if migration is aborted. This is made plausible by looking at each step. Steps 1-4 are analogous to duplication. If migration fails on step 5 the subject is not allowed access to the origin virtual resources in *execute* mode (i.e., ${}^{\gamma}\rho_3$ return the decision *no*). As in step 3, the system is secure independently of whether the *execute* access privilege is set, which means that it is possible to manually set the *execute* access privilege through the cloud management process, to repeat step 5, and to successfully complete the duplication. If migration fails on step 6 then the subject is not allowed to delete the object (i.e., ${}^{\gamma}\rho_{10}$ return the decision *no*). There are two cases: (1) the subject is not allowed to access the object in *control* mode, and (2) there exist other subjects having access to the object. In both cases, the system is secure since no information flow is allowed. In the first case, the subject is allowed to access the object in *write* and *execute* mode but not in *control* mode. This is the case if the object was created by another subject because the *control* access privilege is only set on object creation. A possible solution is that the cloud management process signals the subject which has access privilege to delete the object. Another solution is to define a request element for handing over the *control* of an object to another subject and define an applicable rule. Since there is no information flow when accessing an object in *control* mode, in both solutions the systems is secure. In the second case, the origin object is locked by the access of another subject. In general, the system remains secure if the object is not deleted, since the system is secure before requesting the object's deletion and ${}^{\gamma}\rho_{10}$ is security and $*$-property preserving (in particular, if requests are rejected). Moreover, after migration, all access operations on the origin object have to be redirected to the new object, since the new object replaces the origin object. Therefore, it is reasonable that all access operations on the origin object terminate after the migration is completed. However, it can take some time until all access operations are terminated. For that reason, the cloud management process has to coordinate the termination and redirection of access operations to ensure the origin object's deletion. Further, the point of time, when the origin object is deleted, has no influence on the system's security since access control on the object is decided by the system normally until the object is deleted.

To conclude, information flow control for all operations controlled by the cloud management process can be modelled as the system ${}^{\gamma}\Sigma({}^{\gamma}\mathbb{R}, \mathbb{D}, {}^{\gamma}\overline{W}, {}^{\gamma}z_0)$ with the rules ρ_9^+. Further, the cloud management process is not modelled as a subject of the system but on the access operations of subjects that are instructed by the cloud management process. In particular, the cloud management process interacts with subjects of multiple security classes and, therefore, operates on multiple levels of security. Consequently, the cloud management process has to be trusted in the sense of not introducing covert channels between objects (i.e., virtual resources) and/or subjects (i.e., hardware resources). In this thesis, it is assumed that the cloud management

process is trusted in the sense of not introducing covert channels. A possible approach using non-interference policies [193] in the cloud management process is discussed in the outlook in Section 7.3.

5.3.4.3 Implementing security policies using security classes and security bindings

To express legal requirements and corporate customers' preferences on virtual resource placement in clouds, security polices have to be defined and enforced at the cloud providers. This is done by defining security classes expressing the necessary and provided level of protection of hardware resources and virtual resources. For example, if a corporate customer requests a virtual resource that is processed only in Germany, the assigned location class is ${}^{loc}SC_{DE} \in {}^{loc}\mathbb{SC}$ where $DE \in \mathbb{T}$ is corresponding territory of the German legislation. Further, the ordering of security classes is used to define security policies for allowed information flow. For example, if information is allowed to flow from Europe to Germany but not vice versa then information flow for the corresponding security classes ${}^{loc}SC_{DE}, {}^{loc}SC_{EU} \in {}^{loc}\mathbb{SC}$ is defined ${}^{loc}SC_{EU} \mapsto {}^{loc}SC_{DE}$. By defining security bindings for hardware resources and virtual resources, the applicability of the security policies to hardware resources and virtual resources is specified.

Figure 5.3: Example for security classes on confidentiality, integrity, availability, and location-determination

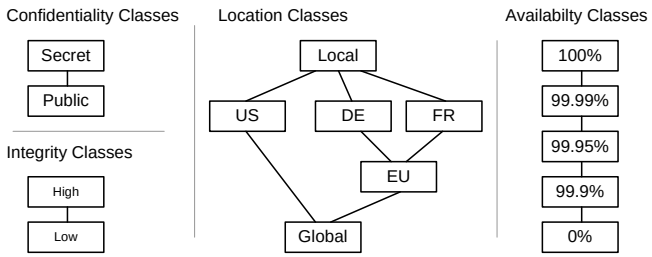


Figure 5.3 depicts an example of security classes with respect to confidentiality, integrity, availability and location-determination. In the figure, information is allowed to flow from bottom to top but not vice versa. Confidentiality is modelled using the two discrete confidentiality classes *public* and *secret*, where, for example, *secret* is applied to resources that support hardware-based encryption (like TPM) and *public* to any resource else. Integrity is also modelled using the two discrete integrity classes *low* and *high*, where, for example, *high* is applied to resources that support cryptographic integrity checks (like remote attestation of TPM) and *low* to any resource else. Location is modelled using the location classes corresponding to the territories of Germany (DE), France (FR), European Union (EU), and USA (US) as well as the lower bound (Global) and upper bound (Local) of location classes ${}^{loc}SC_{global}, {}^{loc}SC_{local} \in {}^{loc}\mathbb{SC}$. Availability is modelled using five availability classes including the lower and upper bound ${}^{av}SC_{0\%}, {}^{av}SC_{100\%} \in {}^{av}\mathbb{SC}$.

Figure 5.4: Example of classifying hardware resources and embedded virtual resources by allowed information flow

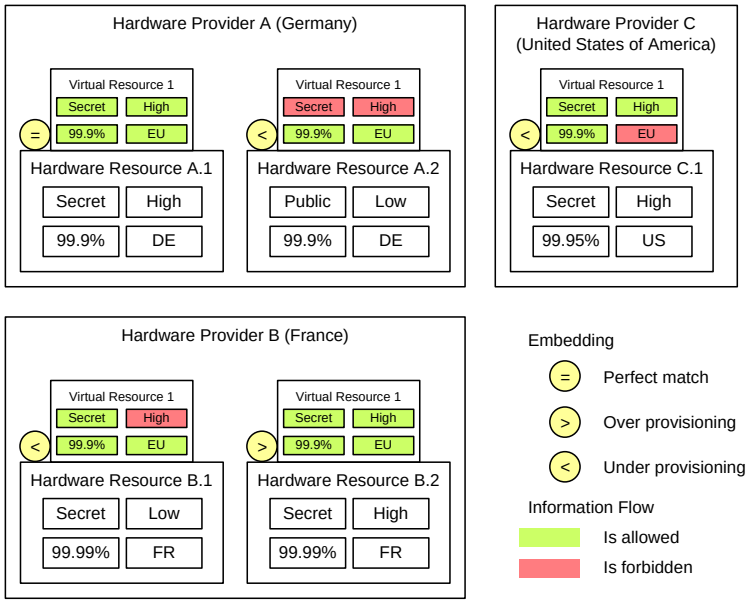


Figure 5.4 depicts an example of applying the security classes depicted in Figure 5.3 to hardware and virtual resources, and deciding on the resource embedding by the cloud provider. There are five hardware resources operated by three hardware providers and one virtual resource processing personal data of German customers. The hardware resources and the virtual resource have assigned security classes and for each hardware resource the placement of the virtual resource is evaluated by allowed information flow. Further, there are two possible placements: (1) on `Hardware Resource A.1` and (2) on `Hardware Resource B.2`. All other hardware resources do not have sufficient security classes. An interesting observation is that `Hardware Resource A.1` is a perfect match and `Hardware Resource B.2` over-provisions in respect of availability. In addition, the under-provisioning of `Hardware Resource B.1` is limited to integrity, which is less severe then the under-provisioning of `Hardware Resource A.2` that also does not fulfil the confidentiality requirements. The `Hardware Resource A.2` under-provisions with respect to location. With respect to data protection law, under-provisioning location is considered a more severe problem than under-provisioning confidentiality or integrity.[1] To efficiently embed virtual resources on hardware resources, it is necessary to avoid over-provisioning. Further, for evaluating compliance breaches (i.e., assigning virtual resources to hardware resources with insufficient security classes) it is necessary to measure the severity of under-provisioning. To address this issue, in Section 5.4.2, metrics on security classes are defined that allow the degree of under- and over-provisioning to be measured.

[1] A transmission to a wrong location is a breach of data protection law since data are transferred to an unauthorised recipient. A lack of confidentiality may result in access to an unauthorised recipient (since data are not sufficiently protected) but does not necessarily have to. A lack of integrity can cause unauthorised modifications but never results in an unauthorised access. However, neglecting confidentiality and integrity is a breach of the due diligence.

During life-time of a virtual resource or hardware resource, it can be necessary to change its security class. In the information model presented in Section 5.3, changing the security classes of objects is allowed, if no subject is accessing it. Further, it is assumed that security classes of subjects do not change. This implies that the security class of a virtual resource can be changed if no hardware resource has access to it, i.e., the virtual resource is neither executed nor stored by any hardware resource. This is reasonable since security classes of virtual resources are changed only on the request of the corporate customers and the changed security classes do not necessarily have to comply with the currently assigned hardware resource. Further, only an entity operating on multiple levels of security is allowed to change the classification, i.e., only the cloud management can do that. Again, this is reasonable since the cloud provider executes the customers' requests (including changing security classes of virtual machines). Consequently, changing security classes of virtual machines is implemented by stopping and withdrawing them from the assigned hardware resource (e.g., by temporarily storing them through the cloud management process and deleting them at the hardware resources). Then, the security class is changed and it is reassigned to a hardware resource with a sufficient security class. That there is no change to the security classes of hardware resources is also reasonable since hardware resources in clouds are not mobile and the security properties of hardware usually do not change. However, it may be necessary to change security classes of hardware resources because of an exceptional event (e.g., relocating servers to another data centre or upgrading availability). This can be done by removing the hardware resource from the system after all access to virtual resources has been terminated and then adding it again with changed security classes. Since all access is terminated before removing the subject, the security of the system is given. However, the cloud management process has to ensure that no covert channels occur (e.g., the hard disk of a server is not deleted before changing security classes).

To conclude, the information model presented in Section 5.3 is applicable to model security policies at the cloud provider, which are necessary to address the challenge of the location inhomogeneity in clouds (cf. Def. 4.6). In particular, (1) the location of data processing is modelled as the location classes of hardware resources, (2) the category of processed data is modelled as the corresponding security classes describing the *necessary level of security*, (3) the origin of processor, controller, and data subject is modelled as the location class of virtual resources which is utilised to process data of the corresponding entity describing the *ensured level of security*, and (4) applicable requirements from contracts and SLA are modelled as security classes with respect to confidentiality, integrity, availability, and location-determination. Therefore, the security classes classify the hardware resources and virtual resources. Security policies describes the information flow allowed between security classes, and they are applied according to the security bindings of hardware resources and virtual resources. Therefore, the lattice-based model on information flow control presented in this section is suitable to address the challenge of the location inhomogeneity in clouds.

5.4 Implementing information flow control

To solve the challenge of location inhomogeneity in clouds (cp. Def. 4.6), it is necessary to describe the context of decision and enforcement. This can be done by using an information

model for a decision based on the *ensured level of security* and to enforce the *necessary level of security*. In Section 5.3, a suitable information model was presented that applies to the cloud management process for deciding and enforcing allowed information flow. In addition to the information model – for addressing the challenge of the location inhomogeneity in clouds – it is necessary to reliably identify information on the location of data processing and its *ensured level of security* as well as to implement decision and enforcement in the resource allocation in cloud infrastructures. Further, monitoring of and reporting on legally compliant decision-making and enforcement is important for fulfilling the contract of IT outsourcing to the cloud (cf. Section 4.3). For that reason, the cloud provider has to monitor the resource allocation and report on its legal compliance to the corporate customer.

In this section, possible approaches on trustworthy resource classification, resource allocation and management, and compliance reporting and management are investigated. First, methods for reliably providing information on the location of hardware resources and their *effective level of security* are discussed. In particular, the management of trust between cloud providers and hardware providers is addressed. Then, implementation of the information model in the resource allocation is investigated with respect to effective decision-making and enforcement. In particular, metrics for measuring over- and under-provisioning with respect to security classes of virtual resources and hardware resources are presented. Finally, methods for monitoring and reporting on compliant resource allocation by the cloud provider are investigated. Particular consideration is given to the application of operation control by the corporate customers.

5.4.1 Trustworthy resource classification

When operating a cloud infrastructure, the cloud provider has to ensure the compliant allocation of hardware resources for hosting virtual resources requested by corporate customers. The decision-making and enforcement when it comes to resource allocation are based on the classification of hardware resources which depends on the hardware resources' location and the *effective level of security* (cf. Section 5.3.4). In general, the hardware resources are not under the direct control of the cloud provider since they are operated by subcontracted hardware providers. Therefore, the cloud provider has to rely on the correctness of the location of hardware resources and the *effective level of security* provided by the hardware provider. In particular, the information on hardware resource location and the *effective level of security* has to be accurate, to allow a correct classification of hardware resources. Since this information is provided by the hardware provider, the cloud provider has to be able to trust this information, i.e., there has to be a relationship of trust between cloud provider and hardware provider. In practice, the relationship of trust is established on a contractual basis. When subcontracting hardware providers, location and the *effective level of security* is defined in the contract signed by the hardware provider. However, provided hardware resources do not necessarily have to comply with the contractual requirements. For example, due to hardware failure the availability of hardware resources is lower that what was contracted. It is also possible that requirements for hardware resources may vary and are requested on demand. For example, hardware resources are highly available only when explicitly requested at the time when virtual resources are assigned, otherwise they have a lower availability. Therefore, for operation control, it is necessary that the cloud provider should have the capability to verify the hardware resource lo-

cation and the *effective level of security*. Manual controls, i.e., inspecting each hosting site and their hardware resources physically, are not suitable for large cloud infrastructures. In particular in global cloud scenarios there are multiple hosting sites established in multiple countries which have to be inspected individually by personnel being physically present. Additionally, hosting sites does not host exclusively for a single cloud provider but can have multiple customers. Thus, physical inspections by each customer are often not practical and, moreover, physical security may be undermined if physical access is frequently granted to externals. In addition, there exist IT security standards for the secure operation of hardware resources (e.g., Trusted Site Infrastructure [206], ISO/IEC 27001 [112]).

A better solution is for all information on location and the *effective level of security* to be provided and validated automatically, and instead of physically inspecting the hosting sites, the hosting sites security is implemented and certified according to IT security standards such as Trusted Site Infrastructure [206] and ISO/IEC 27001 [112]. Consequently, the question arises as to what technical possibilities exist to validate hardware resources' location and an *effective level of security* remotely and automatically. In the context of this thesis, the hardware resources' *effective level of security* depends on their capabilities to ensure confidentiality, integrity, and availability. In the following, technical capabilities when it comes to remote and automated validation of confidentiality, integrity, availability, and location are discussed:

Confidentiality can be ensured remotely by transmitting only encrypted data and keeping the key for decryption secret. Data processing on encrypted data at the current state of the art is possible using homomorphic encryption schemes. They are, however feasible only for simple operations like computing sum and variance of numeric values [147]. Complex operations on encrypted data like program or virtual machine execution are still beyond the state of the art. Therefore, data has to be decrypted at remote locations to enable data processing. For that reason, hardware resources at remote locations have to be able to protect decrypted information from disclosure, and the effectiveness of enforcing non-disclosure has to be attested remotely. Here, it is possible to ensure trusted computing in cloud infrastructures by using a trusted hypervisor [79]. In particular, trusted hypervisors can be utilised to ensure confidential data processing on virtual resources by controlling access to virtual resources and enforcing the encryption of virtual storage [180, pp. 19-26]. Further, trusted hypervisors support remote attestation of executed virtual resources and software [79] [180, pp. 16-19], which can be utilised to validate whether there was access to confidential virtual resources and they were operating correctly. This information can be used to validate whether confidentiality has been ensured at remote locations, and particularly, to detect unauthorised access to virtual resources [80].

Integrity is difficult to ensure remotely since existing methods on integrity protection address detection and correction of modifications but do not prevent the processing of modified data (cf. Section 4.2.3.1). Therefore, hardware resources at remote locations have to be able to protect the integrity of virtual resources (e.g., ECC [94] and cryptographic hashes [59]). Further, it is possible to validate integrity by using cryptographic hash functions in combination with signature schemes reliably (e.g., Digital Signature Algorithm (DSA) [152]). Using trusted hypervisors as well, the integrity of virtual resources, executed software and processed data at remote locations can be attested remotely [79] [180, pp. 16-19].

Availability is also difficult to ensure remotely since controlling availability requires physical access to the hardware resource. Therefore, the availability of hardware resources has to be ensured at the remote location (i.e., by the hardware provider). A possible strategy to address this issue remotely is to introduce redundancy where the use of multiple hardware resources ensures the operation of specific virtual resources (e.g., distributed storage replication [124]). This can help to ensure the availability of virtual resources but does not help to ensure the availability of individual hardware resources. However, the availability of a specific hardware resource can be validated remotely by measuring its uptime and downtime based on possible remote access and calculating the asymptotic availability (cf. Def. 5.43).

Location of hardware resources cannot be ensured remotely since physical control of the hardware resource is necessary. Further, determining the location of hardware resources remotely is a widely investigated problem. In wired networks, existing approaches can be classified into semantic-based and measurement-based. In semantic-based approaches, the information on the IP address is mapped to location information, for example by using the routing information and location information of reference hosts to locate clusters of IP addresses [160]. In measurement-based approaches, the round trip time is used to estimate the propagation delay and the resulting distance to so called 'landmarks' (i.e., hosts with known location), for example, *Shortest Ping* [226]. There also exist approaches using multiple landmarks, e.g., *Constraint Based Geolocation (CBG)* [91]. Further, there exist hybrid approaches which are semantic- and measurement-based, for example, *3-Tier Geolocation* [218] which estimates the postal address of hosts by combining *Constraint Based Geolocation*, *traceroute*, and mapping IP addresses to postal codes. Semantic-based approaches are limited by the topographic neighbourhood of the IP addresses and do not apply to private address spaces.[1] In virtual networks, private address spaces are used, and at the access network, gateways with Network Address Translation (NAT) are usually used, masking the connection endpoint in the virtual network. Measurement-based approaches are limited by the accuracy of measuring the propagation delay. Since the round-trip time is measured to estimate the propagation delay, the measurement is influenced by any delay that applies to the round-trip time. In virtual networks, the substrate network does not necessarily have the same topology as the embedded network. Therefore, additional delays are introduced by hidden routers and links in the substrate network. Further, the connection end-point does not necessarily have to be the host processing the data, since network relaying to the data processing host is possible. Then, the propagation delay after the connection end-point is indistinguishable from the processing delay of the back-end system. Nonetheless, measurement-based approaches can be accurate if sufficient landmarks are available and they are combined with semantic-based approaches – for example, *3-Tier Geolocation* has a mean error of less than 1 km [218]. An alternative approach to determining the location of hardware resources remotely is to let the hardware resources themselves provide location information and validate the correctness of this information (e.g., by plausibility checks using measurement-based location determination like *3-Tier Geolocation*). If hardware resources support TPM it is possible to authenticate hardware resources uniquely by their cryptographic key [15]. In addition, the location of hardware resources with TPM support can be verified through physical inspection (particularly by authorised third parties), and it is possible

[1] According to RFC 1918 [169], private address spaces are `10.0.0.0/8`, `172.16.0.0/12`, and `192.168.0.0/16`.

to reference these hardware resources in contracts.

To conclude, cloud providers can classify hardware resources' thrustworthiness by confidentiality, integrity, availability, and location, since it is technically possible to verify the classification remotely for each hardware resource. The key technology is the hardware resources' support of TPM which enables remote attention and authentication of hardware resources.

5.4.2 Resource allocation and management

The compliant allocation of hardware resources for hosting virtual resources requires the consideration of security bindings of hardware resources and virtual resources and the allowed information flow between them. This has a major impact on resource allocation and management since the allowed embeddings of virtual resources on hardware resources are limited by security constraints. For practical implementation, it is necessary to provide measures to establish whether an embedding is compliant with the security constraints, and for optimal resource utilisation, the degree of over-provisioned hardware resources in respect of security constraints of virtual resources has to be quantified. Based on the qualitative observations in Section 5.3.4.3, a quantitative approach to validating the resource embedding and its integration into existing resource allocation strategies are discussed in this section. First, a brief overview of the resource allocation problem and the importance of metrics on optimisations criteria is given. Then, possible approaches to and application of quantitative metrics based on security classes are discussed.

In general, resource allocation is an NP-hard optimisation problem which corresponds to the multi-dimensional bin packing problem (e.g., finding an embedding of all virtual resources [214]), to the knapsack problem (e.g., maximising the economic profit and SLA fulfilment [210]), and to the min-max optimisation problem generally (e.g., minimising required hardware resources and maximising the use of hardware resources [133]). In the context of virtual network embedding, a comprehensive classification of the resource allocation problem and its possible solutions is provided by Fischer et al. [73]. In the survey, it is observed that possible solutions for virtual network embedding can be classified into accurate and heuristic solutions. Moreover, the quality of the heuristic solutions is evaluated by using metrics based on optimisation criteria like QoS-compliance, economic profit, and resilience. These observations also apply to resource allocation in clouds where existing approaches address similar optimisation criteria (e.g., performance, costs, locality, and reliability) on recource allocation [132, pp. 12–16]. Consequently, metrics on optimisation criteria are an important input for resource allocation strategies and their evaluation. This is also true when implementing resource allocation based on security bindings and allowed information flow. Metrics with respect to confidentiality, integrity, availability, and location are required to evaluate resource allocation according to the constraints of information flow control.

A good candidate for defining metrics are the distances between the partially ordered security classes. By comparing the required security class with the assigned security class, it is then possible to measure the quality of the embedding. Because of the partial order, there are four cases that have to be considered when comparing the required security class $SC_{req} \in \mathbb{SC}$ with the assigned security class $SC_{ass} \in \mathbb{SC}$:

Case 1 ($SC_{req} = SC_{ass}$): If both security classes are equal there is a *perfect match* and the distance is defined as zero.

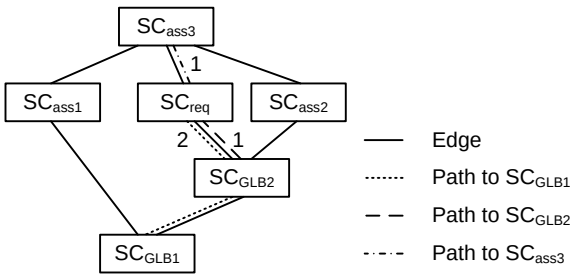
Case 2 ($SC_{req} \mapsto SC_{ass}$): If information flow is allowed from the required security class to the assigned security class then there is a *valid match*. If the two security classes are not the same there is over-provisioning and, assuming the security classes are ordered equidistantly, the distance between two security classes is measured by counting the number of edges on the path between the two security classes within the lattice of security classes.[1]

Case 3 ($SC_{ass} \mapsto SC_{req}$): If information flow is allowed from the assigned security class to the required security class then there is an *invalid match*, if the two security classes are not the same. Then, there is under-provisioning and the distance is defined by the negative distance of both security classes analogously to case 2.

Case 4 (No information allowed): If there is no information flow allowed between required and assigned security classes then there is an *invalid match*. Since there is no information flow allowed, the distance cannot be measured by the distance between both security classes. However, the greatest lower bound of both security classes defines the smallest security class to where information is allowed to flow from both security classes and a valid match would have been possible. This makes the greatest lower bound of both security classes a good candidate to specify the distance of both security classes. Thus, the distance is defined (analogously to case 2 and 3) by the negative distance from the required security classes to the least upper bound of both security classes.

Figure 5.5 exemplifies the paths in the lattice of security classes for a single required security class $SC_{req} \in \mathbb{SC}$ and three assigned security classes $SC_{ass1}, SC_{ass2}, SC_{ass3} \in \mathbb{SC}$. The information flow is allowed from bottom to top. The information flow from SC_{req} to SC_{ass1} and SC_{ass2} is not allowed. Therefore, the distance is calculated by the path from SC_{req} to the corresponding greatest lower bound which is $SC_{GLB1} = SC_{req} \oplus SC_{ass1}$ and $SC_{GLB2} = SC_{req} \oplus SC_{ass2}$, respectively. The calculated path has a length of 2 for SC_{GLB1} and of 1 for SC_{GLB2}, respectively. Since the information flow is not allowed, the corresponding distance calculated by the metric is negative. Further, the information flow from SC_{req} to SC_{ass3} is allowed and the calculated path has a length of 1. Since the information flow is allowed, the corresponding distance calculated by the metric is positive.

Figure 5.5: Example of paths and their lengths in the lattice of six security classes for a single required and three assigned security classes. Information flow is allowed from bottom to top.



[1] Edge and path are used in the context of the graph theory, i.e., an edge is a link between two vertexes (i.e., security classes) and a path is a sequence of edges which connects a sequence of vertices.

According to the previous considerations, the metric $d_{SC}(SC_{req}, SC_{ass})$ measuring the distance between required security class SC_{req} and assigned security class SC_{ass} is defined as follows.

$$d_{SC}(SC_{req}, SC_{ass}) = \begin{cases} 0 & : \; SC_{req} = SC_{ass} \\ \left|\left\{SC \in \mathbb{SC} \backslash \{SC_{req}\} : SC_{req} \mapsto SC \wedge SC \mapsto SC_{ass}\right\}\right| & : \; SC_{req} \mapsto SC_{ass} \wedge SC_{req} \neq SC_{ass} \\ -\left|\left\{SC \in \mathbb{SC} \backslash \{SC_{req}\} : SC_{ass} \mapsto SC \wedge SC \mapsto SC_{req}\right\}\right| & : \; SC_{ass} \mapsto SC_{req} \wedge SC_{req} \neq SC_{ass} \\ -\left|\left\{SC \in \mathbb{SC} \backslash \{SC_{req}\} : (SC_{req} \oplus SC_{ass}) \mapsto SC \wedge SC \mapsto SC_{req}\right\}\right| & : \; else \end{cases}$$

where $SC_{req}, SC_{ass} \in \mathbb{SC}$ security classes. (5.1)

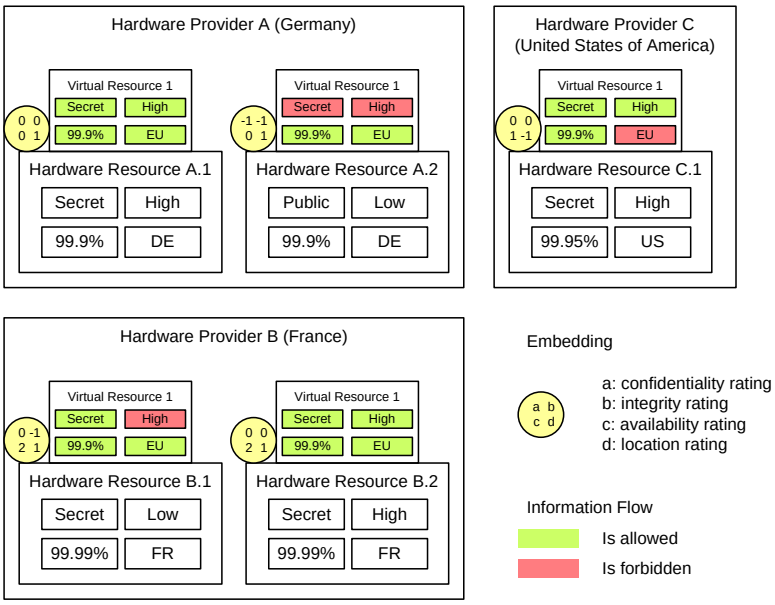
The number of edges in the lattice is equal to the number of security classes on the path minus one. Therefore, the security classes on the path are counted without starting security class (or ending security class, respectively) SC_{req}. If the required and the assigned security classes are not equal, the number of security classes in $\mathbb{SC} \backslash \{SC_{req}\}$ where information flow is allowed as specified in Equation 5.1 are counted, which is equal to the number of security classes on the path minus one. This can be made plausible when looking at Figure 5.5. For example, the path from SC_{req} to SC_{GLB1} has the length 2. The security classes $SC \in \mathbb{SC} \backslash \{SC_{req}\}$ where information flow is allowed to flow from SC_{GLB2} to SC are $\{SC_{GLB1}, SC_{GLB2}\}$. Further, the security classes $SC \in \mathbb{SC} \backslash \{SC_{req}\}$ where information flow is allowed to flow from SC to SC_{req} are $\{SC_{GLB1}, SC_{GLB2}, SC_{ass3}\}$. The intersection of both results is $\{SC_{GLB1}, SC_{GLB2}\}$ which has a cardinality of 2.

Figure 5.6 illustrates the application of the metric d_{SC} for the quantitative evaluation on an example[1] of virtual resource embedding. In the example, the metric is applied on confidentiality classes, integrity classes, availability classes and location classes individually for each hardware resource. The assignment of VIRTUAL RESOURCE 1 to HARDWARE RESOURCE A.1 returns a perfect match for the distance of the confidentiality classes, integrity classes, and availability classes, since required and assigned security classes are equal. For location, the required security class is SC_{EU} and the assigned security class is SC_{DE}. Here, case 2 of the metric applies since information flow is allowed. The calculated distance is 1 since SC_{EU} and SC_{DE} are directly connected in the lattice (cf. location classes illustrated in Figure 5.3). The assignment of VIRTUAL RESOURCE 1 to HARDWARE RESOURCE C.1 has a negative distance for location classes since case 4 of the metric applies and the greatest lower bound SC_{Global} is used for calculating the path length resulting in a negative distance -1.

The metric d_{SC} can be used in resource allocation strategies to find optimal embeddings having a distance close to zero. To compare embeddings on the basis of different dimensions, summing up non-negative values provides good results. However, considering negative values is more complex, since summing up can result in distances close to zero, while the distance on

[1] The presented example corresponds to the example used in Figure 5.4. The corresponding security classes of the example are illustrated in Figure 5.3.

Figure 5.6: Example for quantitative evaluation of virtual resource embedding.



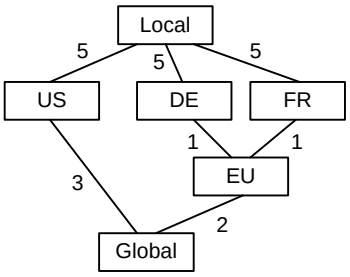
each dimension can be high but positive and negative distances annihilate each other. A possible solution is to sum up the absolute values, and if negative values (due to under-provisioning) should be avoided then it is possible to sum up positive and negative values separately. The latter allows the optimisation for the embedding explicitly in respect to avoiding negative values.

Further, the metric d_{SC} is based on the assumption that all security classes are equidistant. This is not necessarily always true. For example, embedding virtual resources with required location class SC_{EU} on a hardware resource with assigned security class SC_{US} can have a more severe impact on legal compliance than embedding virtual resources with required location class SC_{DE} on a hardware resource with assigned security class SC_{FR}. This results from the fact that Germany and France are both in the legislation of the EU/EEA, which is an area with harmonised legislation, while the USA and the EU/EEA do not have harmonised legislation. For that reason, a weighted distance model can be more suitable in providing more accurate results for the evaluation of embedding quality. In addition, every edge in the lattice is weighted with a factor $\rho \in \mathbb{R}^+$ and the length of a path in the lattice is calculated by summing all weights of the path's edges.

Figure 5.7 illustrates an example of weighted distances of location classes. It is assumed that location classes corresponding to a country are on the same level. Based on that, the edges from location classes corresponding to a country to the maximum location class SC_{Local} is weighted highest, with 5 since placement in the cloud or locally considered the decision with the most severe impact. The edges between location classes corresponding to a country and the minimum location class SC_{Global} is weighted with 3 since it is still a severe decision to place virtual resources within a specific country or globally. The edges between location classes corresponding to a member state of the EU/EEA and the location class corresponding to the EU/EEA are weighted with 1 and the edge between location classes corresponding to the

EU/EEA and SC_{Global} is weighted with 2. Then, the sum of weights on the path from location classes corresponding to a member state to SC_{Global} is 3, which corresponds to the weight of edges between location classes corresponding to a country and SC_{Global}. Further, the placement within the EU/EEA is considered to have less severe impact than the placement outside of the EU/EEA.

Figure 5.7: Example of weighted distances of location classes.



The weights can be assigned for each edge in the lattice individually and with respect to applicable security requirements. It is also possible to use weights to compare security classes of different security requirements. For example, edges of location classes are weighted at ten times the upper bound of distances between integrity classes to ensure that location has a higher priority than integrity when embedding virtual resources.

To conclude, the lattice-based model for information flow control also supports the definition of metrics which can be used for evaluating the quality of resource allocation in clouds with respect to confidentiality, integrity, availability, and location. The metrics are defined based on the length of paths between security classes in the lattices. The distance of incomparable security classes can be described by the distance to the greatest upper bound. Moreover, distances can be measured either based on equidistant security classes or on weighted edges in the lattice.

5.4.3 Compliance monitoring and reporting

In the scenario of IT outsourcing to the cloud, compliance management by the cloud provider and reporting on compliance to the corporate customers and authorised third parties is necessary (cf. Section 4.3). When applying lattice-based information flow control to resource allocation in clouds, the cloud provider needs mechanisms to monitor the effective resource allocation and its compliance with the applied information control policies. Further, the results of compliance monitoring have to be reported to the corporate customer. In the following, compliance monitoring based on logging and documentation on the hardware resource and cloud management level is investigated. Then, the reporting on compliant resource allocation by the cloud provider to the corporate customer and authorised third parties is discussed.

Compliance monitoring in cloud infrastructures

Compliance monitoring requires logging and documentation of the performed actions and can

be done in clouds at the level of hardware resources, virtual resources, and the cloud management (cf. Section 4.3.1). The cloud provider operates the cloud management and, therefore, can access logging data and documentation on the cloud management level directly. The hardware resources are operated by the hardware provider. Consequently, the cloud provider has to request the necessary logging data and documentation from the subcontracted hardware providers. In Section 5.4.1, methods for verifying the security classification of hardware resources remotely by the cloud provider are identified. These methods apply to monitoring the compliance of resource allocation with information flow control policies and can also be used to collect logging data and documentation provided by hardware resources. When looking at IaaS, the virtual resource level is under the control of corporate customers and is not available to the cloud provider (cf. Section 4.3.1). However, the corporate customer may use logging information from the the virtual resource level to perform plausibility checks of the compliance reports of the cloud provider. Further, logging data and documentation from the virtual resource level are not necessarily required for monitoring compliant resource allocation by cloud providers if logging data and documentation from the hardware resource level and cloud management level are available. This is because the resource allocation has to comply with the allowed information flow of virtual resources, and therefore depends on the correct assignment of virtual resources to hardware resources with respect to their security classification. The security classification of hardware resources is monitored on the hardware resource level. The correct assignment of virtual resources to hardware resources according to their security classification is done in the cloud management process, and is thus monitored on the cloud management level. Therefore, the cloud provider has access to logging data and documentation that is sufficient to monitor the compliance of resource allocation in clouds.

A possible approach is to use self-describing hardware resources in combination with TPM support, providing trustworthy information on their security classification (cf. Section 5.4.1). In particular, the TPM support allows the implementation of trusted hypervisors that can be used by the cloud provider to test the integrity of virtual resources remotely [79]. The security classification of virtual resources, the security policies on allowed information flow, and the decisions on resource allocation can be logged and documented in the cloud management process. If the cloud management process is operated on TPM-supporting hardware resources, the integrity of the cloud management process can be attested, too. The compliant execution of the cloud management process can be evaluated by using security metrics as described in Section 5.4.2 if logging and documentation covers reliable identification of hardware and virtual resources, their security classification and accurate information of time and mode of access to virtual resources by hardware resources. The results of the evaluation can be used by the cloud provider to monitor the compliance of the resource allocation at run-time and, if the results of the evaluation are stored, also at any point in the future.

Compliance reporting to customers and authorised third parties

It is possible to use the evaluation results of the compliance monitoring to generate reports for corporate customers (e.g., for the purpose of service operations controls) and authorised third parties (e.g., for the purpose of IT security auditing). The metrics on confidentiality, integrity, availability, and location which were discussed in Section 5.4.2 can be used to automatically

detect SLA violations and, therefore, are applicable to extending existing approaches on automated compliance reporting based on SLA violations [25] [63]. The evaluation results of the compliance monitoring can also be used as input for reporting on service operation controls. For example, when reporting on service operation controls by ISAE 3402, the [107] the auditor inspects the evaluation results of the compliance monitoring with respect to virtual resources utilised by the corporate customer.

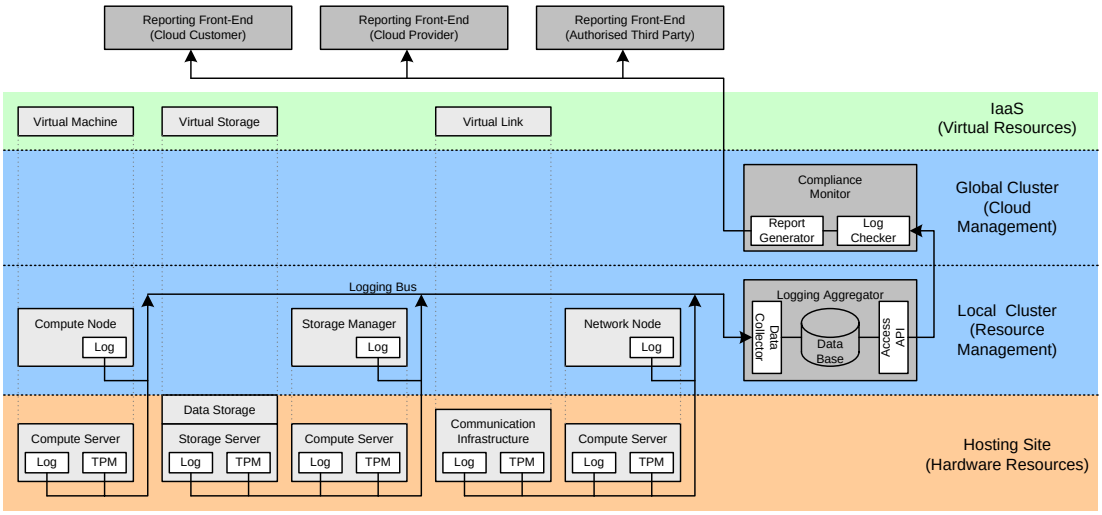
In general, it is important to consider that externals usually do not have knowledge of the cloud infrastructure and allocated hardware resources the way the cloud provider does. In particular, the externals' interests lie on verifying the compliance of utilised virtual resources. Therefore, information on the cloud infrastructure and allocated hardware resources is required only when the SLA violation is analysed or the provided information in the compliance report is verified for its correctness. In addition, it is generally not in the interest of the cloud provider to provide internal information on the cloud infrastructure and hardware resources unnecessarily (e.g., due to protecting trade secrets). Therefore, compliance reports to externals should focus in general on the utilised virtual resources and their compliance with respect to requested location and the *effective level of security*. Additional information on how and where to find information on involved hardware resources and related logging data should be provided to support the analysis and verification of the information provided in the report. This can be covered by providing contact information for the entity that is responsible for hosting the virtual resource (i.e., usually the cloud provider but it can also be the responsible hardware provider) and the hardware resource identifiers (e.g., the cryptographic key to TPM-supporting hardware resources [15]) to link the information provided in the report with corresponding logging data and documentation at the cloud provider or hardware provider.

Monitoring and reporting architecture in cloud infrastructures

Assuming that all hardware resources support TPM, it is possible to attest to the security classification of hardware resources remotely (cf. Section 5.4.1) and access operations performed by hardware resources (cf. *compliance monitoring in cloud infrastructures* in this section). In particular, it is possible to use the logging data to evaluate the quality of resource allocation (cf. Section 5.4.2) and detect SLA violations (cf. *compliance reporting to customers and authorised third parties* in this section). This can be utilised to build a monitoring and reporting architecture that aggregates and checks logging data and generates compliance reports automatically.

Figure 5.8 illustrates a possible architecture for compliance monitoring and reporting in cloud infrastructures. For each local cluster, logging data and results of the TPM-remote-attestation are collected via a logging bus and stored in a database at the LOGGING AGGREGATION. The LOGGING AGGREGATION provides via an access API the data to the global cluster's COMPLIANCE MONITOR which performs automated log checking and report generation. The logs are checked by verifying the security classification of hardware resources and applying metrics for evaluating the quality of the resource allocation (cf. Section 5.4.2). The reports are then generated based on the evaluation results specifically for cloud customers, cloud providers, and authorised third parties. The logging aggregation is located at the local cluster since the local cluster directly interacts with the hardware provider and, therefore, with

Figure 5.8: Monitoring and reporting architecture in cloud infrastructures.



the hardware resources. The local clusters are coordinated by the global clusters which perform the cloud management operations. Therefore, the compliance monitoring is located at the global clusters since there the decisions of resource allocation have to be validated with respect to the corporate customers' requests.

For a secure and trustworthy operation of the monitoring and reporting architecture, it is necessary to ensure the integrity and authenticity of the logging data and reports with respect to their communication and their processing by the LOGGING AGGREGATOR and COMPLIANCE MONITOR. Further, access control mechanisms have to be implemented at the reporting front-ends to ensure authorised access only. In addition, the information provided in the reports has to specifically address the purpose of the reporting and must not contain information that is not to be disclosed to the recipient of the report (e.g., reports to corporate customers must not contain data correlated to other corporate customers).

In Section 6.1, a proof-of-concept implementation of the proposed architecture based on the OpenStack cloud platform is presented and evaluated for the scenario of compliance monitoring and reporting with respect to location-determination.

5.5 Conclusions on tackling location inhomogeneity in clouds

The work presented in this chapter shows how the challenge of location inhomogeneity in clouds can be addressed by controlling the flow of information between virtual resources. The results are as follows.

1. A **classification of information flows in clouds** and their interdependency identifying the *information flow of processed data*, which is under control of corporate customers, and *information flow of virtual resources*, which is under control of cloud providers (cf. Section 5.1)).

2. A **lattice-based model for information flow control on virtual resources** combing and extending existing methods of Bell and La Padula, Denning, and Biba to support multidimensional security classifications generally (cf. Section 5.3).

3. Two **lattices of security classes modelling availability and location constraints**, which are used (in conjunction with existing lattices on confidentiality and integrity) to model information flow control in clouds tackling the challenge of location inhomogeneity (cf. Section 5.3.2 and Section 5.3.3).

4. A proposal for the **implementation of information flow control in the cloud management process** including a trustworthy resources classification, secure resource allocation and management, and reliable compliance monitoring and reporting (cf. Section 5.4).

5. A **security-class-based metric for evaluating the quality of resource allocation** in clouds, which allows to compare mutli-dimensional security characteristics and is ready to use in existing resource allocation algorithms (cf. Section 5.4.2).

6. A **compliance monitoring and reporting architecture** in clouds, which is based on the classification of virtual resources and hardware resources (cf. Section 5.4.3).

It is now possible to introduce information flow control to the cloud management process, which is able to allocate cloud resources according to the *effective levels of security* at the hardware resource locations. The corporate customers can assign the *necessary level of security* when requesting virtual resources, which are then classified to the corporate customers' specifications. The methods of information flow control enable the cloud provider to properly assign the requested virtual resources to hardware resources in compliance with the *necessary level of security*. Further, the trustworthy classification of cloud resources allows for continuous monitoring of resource placement and enables compliance reporting to corporate customers and authorised third parties. Moreover, the cloud provider can use the monitoring and reporting for internal inspections. In Chapter 6, the feasibility of the approach presented is investigated through experimental evaluation of a proof-of-concept implementation and analytical evaluation in respect to technical and legal characteristics.

Chapter 6

Implementation and Evaluation

In Chapter 5, an information model is presented which is able to describe information flow control based on legal requirements identified in Chapter 3. A particular requirements are access and transfer control of cloud resources, whose technical implementation bears the challenge of location inhomogeneity (cf. Def. 4.6). This challenge is addressed in this thesis particularly by introducing location-determined data processing in clouds.

In this chapter, a proof-of-concept implementation of location-determined data processing including compliance monitoring and reporting is presented (Objective 6). Further, the feasibility of trustworthy compliance management in clouds is evaluated experimentally on the basis of the implemented demonstrator in OpenStack and analytically with respect to scalability, reliability, trustworthiness, and legal compliance (Objective 7). In conclusion, the legal and technical boundary implicated by established legislation and the cloud computing paradigm (Objective 4) is identified. In particular, the boundary of tackling the challenge of location inhomogeneity in clouds (cf. Def. 4.6) is determined.

In Section 6.1, the implementation of the approach proposed in Section 5 is described and evaluated experimentally with respect to location-determined data processing in OpenStack. For that reason, the resource management and logging mechanisms of OpenStack are introduced (cf. Section 6.1.1) and their extension to the Location-Determining resource management and logging Architecture (LDA) developed in this thesis is described (cf. Section 6.1.2). Further, the proof-of-concept experiment and its results are presented (cf. Section 6.1.3). In Section 6.2, the trustworthiness and legal compliance of cloud computing is evaluated with respect to observations made in the experiment and with respect to legal and technical considerations in general. First, the achieved legal compliance and the trade-off between legal compliance and usability (cf. Section 6.2.1) are investigated. Further, the technical feasibility of legally compliant clouds is investigated (cf. Section 6.2.2). In particular, technical observations on scalability, reliability, and trustworthiness made in the experiment are discussed. In conclusion, the legal and technical boundaries of tackling the challenge of location inhomogeneity in clouds are identified in Section 6.3.

6.1 Implementing and evaluating location determination in OpenStack

As a proof of concept and to evaluate the methods for information flow control in clouds that were presented in Chapter 5, location-determining resource management and logging is implemented in the IaaS cloud platform OpenStack [157]. Location determination is selected for the proof of concept, because it provides a complex lattice of security classes with incomparable security classes, and from the legal perspective, it is considered a fundamental security criterion for achieving legal compliance (cf. Section 3.6.1). Confidentiality, integrity, and availability can be addressed the same way, but require different input when verifying the correctness of the security classification of hardware resources (cf. Section 5.4.1).

OpenStack [157] is selected as reference platform for implementing the proof of concept for multiple reasons. First, OpenStack is a prominent cloud platform in the literature and in praxis (cf. Section 4.1.2.1). Further, the OpenStack infrastructure is representative for IaaS cloud infrastructures since it implements all components of a cloud infrastructure in the IaaS cloud computing ontology (cf. Section 4.1.4). In addition, OpenStack supports the pooling of hardware resources which can be utilised for location-determined resource allocation, and moreover, sophisticated and extendible logging of resource allocation is also supported by OpenStack (cf. Section 6.1.1). Last but not least, OpenStack is fully open source and documented [157]. Therefore, modification and extension of existing program code is possible. Consequently, OpenStack is a good candidate for implementing a proof of concept.

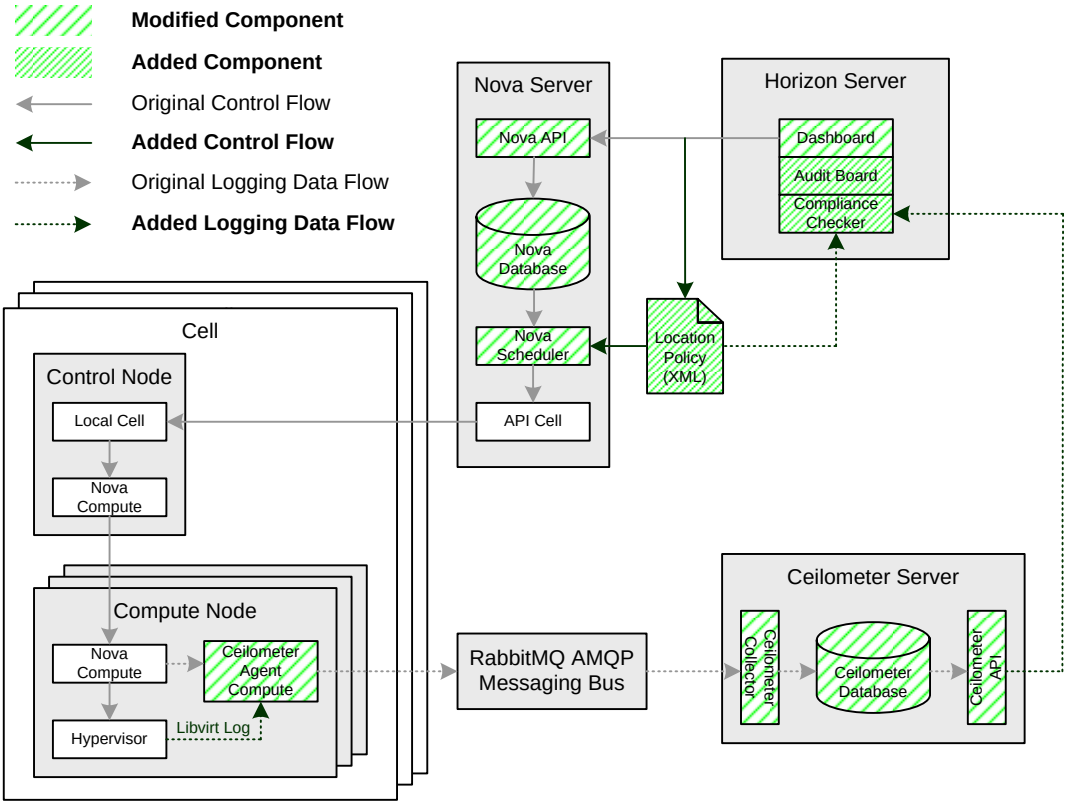
In this section, the implementation and evaluation of the LDA in OpenStack is presented. First, the resource management and logging in OpenStack and their extensions to the LDA implemented in this thesis are described. Then, the experiment and results are presented.

6.1.1 Resource management and logging in OpenStack

The resource management and logging in OpenStack is visualised in Figure 6.1. The figure also includes the extensions necessary for implementing the LDA. In the following, the original components of OpenStack are introduced briefly, providing a basic understanding of the extensions, which are then presented in Section 6.1.2.

In OpenStack, hardware resources are organised by *OpenStack Compute Cells* (cf. 'cells' in OpenStack documenation [157]). Introduced in the OpenStack Version 'Grizzly' in 2013, these cells are considered experimental and are not included in the stable releases before the OpenStack Version 'Juno' in October 2014 (which is after the completion of implementation and experimentation). For the purpose of this thesis, the progress of development in the version from 2013 is mature enough to fully demonstrate location-determined resource allocation based on cells. Cells are organised hierarchically on multiple levels and represent local clusters in the cloud infrastructure (cf. Section 4.1.4). Each cell consists of a *control node* (cf. 'compute manager' in Section 4.1.4) responsible for the management of the cell's hardware resources and *compute nodes* hosting the hypervisors running corporate customers' virtual machines. Cells are controlled via the so-called API CELL encapsulating the management of each cell. When requesting a virtual machine to be spawned on hardware resources of a specific cell,

Figure 6.1: Extended resource management and logging architecture for virtual machine provisioning in OpenStack.



the API CELL is called with parameters of the requested virtual machine and the target cell. The request is then forwarded to the target cell and managed locally by the target cell's control node.

The resource management is performed by the module *OpenStack Compute Nova* (cf. 'compute' in OpenStack documenation [157]). The module consists of the *Nova Server* coordinating the resource management and service orchestration (cf. 'compute fabric' and 'service orchestrator' in Section 4.1.4) and local NOVA COMPUTE agents located at each control node and each compute node. The *nova server* is contacted via the NOVA API accepting virtual resource requests and returning status information on the operated virtual resources. All status and control information of operated virtual resources is stored in the NOVA DATABASE. The allocation of hardware resources and the placement of virtual resources is performed by the NOVA SCHEDULER. Based on the parameters provided to the NOVA API, hardware resources are selected and the configuration of the virtual resource is forwarded via the API CELL to the local agent NOVA COMPUTE at the control node and the executing compute node.

The management front-end in OpenStack is provided by *OpenStack Dashboard Horizon* (cf. 'dashboard' in OpenStack documenation [157]). The DASHBOARD is operated on the *Horizon Server* and provides management access for cloud providers and corporate customers.

Access to the DASHBOARD is granted via the web using HTTP and HTML. There are different views for the cloud provider, i.e., a*dmin view*, and the corporate customers, i.e., *user view*. In admin view, the DASHBOARD provides an overview on all hardware resources and all virtual resources. In user view, only the user's virtual resources are visible. In addition, the DASHBOARD allows in both views to manage virtual resources, and particularly, to request, configure, and release them.

Logging services of virtual resources is provided by *OpenStack Telemetry Ceilometer* (cf. 'telemetry' in OpenStack documenation [157]). The logging architecture consists of the logging aggregator operated on the *Ceilometer Server* and the local agents CEILOMETER AGENT COMPUTE located at each compute node. In the original version, the local agents collect information on the actions executed by the local NOVA COMPUTE and provides the information via the *RabbitMQ AMQP Messaging Bus* (cf. 'RabbitMQ' in OpenStack documenation [157]) for collection. The logging data are then collected by the CEILOMETER COLLECTOR and stored in the CEILOMETER DATABASE. The logging data can be accessed by using the CEILOMETER API providing the information via SQL requests sent to the CEILOMETER DATABASE and forwarding the SQL answers in return. All logging data are handled in OpenStack as key-value pairs which are organised into JSON objects [66]. By default, there is no visualisation service for logging data.

6.1.2 Location-determining resource management and logging architecture

When addressing the challenge of location inhomogeneity in clouds (cf. Def. 4.6) with respect to location determination in OpenStack, it is necessary to extend (1) resource management to include the ability to decide and enforce the location of virtual resource placement and (2) the logging mechanisms to provide all information necessary for compliance monitoring and reporting. This extension is called the Location-Determining resource management and logging Architecture (LDA), and is presented in the following.

The implementation work presented in this thesis was developed between 2012 and 014 with the support of several bachelor's and master's theses [19] [118] [23] [58], which were (co-)supervised by the author of this thesis.

In parallel to this thesis, the concept of *geo-tagging* using *trusted computing pools* was developed in OpenStack by the Intel Corporation and demonstrated first on 11 April 2013 at the Forum of the National Cybersecurity Center of Excellence (NCCoE) [217]. The concept of *geo-tagging* is based on the draft on *Trusted Geolocation in the Cloud: Proof of Concept Implementation* developed by the NIST since December 2012 [17]. Correspondingly, there exists an OpenStack blueprint on *geo-tagging*[1] which was created[2] on the 25th of September 2013 and proposes generally the same code intervention points in *OpenStack Compute Nova* and *OpenStack Dashboard Horizon* as were chosen in this thesis. The implementation of the OpenStack blue-print is not publicly available and not yet included in any released version of

[1] The OpenStack blue-print on *geo-tagging* can be found in the OpenStack Wiki at https://wiki.openstack.org/wiki/GeoTagging (last visited: 30.06.2015).

[2] According to the page information of the OpenStack wiki the blueprint was created on 09:29, 25. Sep. 2013 (in the OpenStack Wiki https://wiki.openstack.org/w/index.php?title=GeoTagging&action=info; last visted: 30.06.2015).

OpenStack.[1] Therefore, a comparison with the implementation work presented in this thesis is limited to the OpenStack blueprint and the correlated demonstration of the *Trusted Geolocation in the Cloud: Proof of Concept Implementation* [217].

The LDA presented in this thesis has been implemented independently of any work done on the concept of *geo-tagging*. Consequently, both implementations are individual pieces of work with their own contribution to a proof of concept for trusted location determination in clouds. In particular, the model for information flow control presented in Section 5.3 marks a major difference between the two implementations since these methods for information flow control are neither included nor considered in the concept of *geo-tagging*. Another major difference between the two implementations is that LDA is based on OpenStack Compute cells while the approach on *geo-tagging* uses *trusted computing pools*[2] which are based on Intel's Trusted Execution Technology (TXT) technology for implementing TPM support and is therefore limited to hardware resources providing TXT support (i.e., using Intel-CPUs). This restriction to a specific hardware vendor (i.e., the Intel Corporation) limits the general applicability to hardware resources of other hardware vendors. The LDA is not limited to hardware resources of a specific hardware provider and can be applied generally.

To implement the LDA, the resource management, logging mechanisms, and the dashboard has been extended by (1) *location-determined resource management*, (2) *location determined logging*, and (3) *compliance reporting*. Figure 6.1 illustrates the components modified and added here as well as additional control flow and logging data flow. Examples of configuration, execution, and logging are presented in the context of the experimental evaluation in Section 6.1.3.

The implementation of **location-determined resource management** consists of (i) adding *location parameters* to the configuration of virtual machines in the DASHBOARD, (ii) extending the NOVA API and NOVA DATABASE to process the added location parameters, (iii) implementing an XML-based location policy determining virtual machine placement with respect to location parameters, and (iv) implementing a location filter in the NOVA SCHEDULER enforcing the location policy [19]. The **location parameters** for virtual machine configuration are *data type* (i.e., category of processed data) and *origin* (i.e., origin of processor, controller, and data subjects or 'data owner') which are required for deciding on the necessary level of protection (cf. Section 3.6.1 and Def. 4.6). In addition, an option for creating a back-up instance of a virtual machine is provided. When selected, two virtual machines are created, one productive virtual machine and one back-up instance. The **location policy** contains rules which return applicable cells in OpenStack based on *data type* and *origin*. An XML Schema Definition (XSD) for XML-based location policies and the example polices used in the experimental evaluation can be found in Appendix C. Assuming that each cell is located in a single specified and known country, the rules of the location policy enables the resource scheduler to decide on the target location of virtual resources based on the parameters, *data type* and *origin*, specified by the corporate customers. Further, a dedicated set of rules for allocating hardware resources for back-up instances is specified which is used when creating back-up instances. The **location**

[1] Last time verified on 30.06.2015.

[2] The OpenStack blueprint on *trusted computing pools* can be found in the OpenStack Wiki on the Internet: `https://wiki.openstack.org/wiki/TrustedComputingPools` (last visited: 30.06.2015).

filter at the NOVA SCHEDULER filters the available cells in OpenStack based on the location parameters and the applicable rules of the location policy. For each requested virtual machine, logs of the applied rules and filtered cells are written at the *Nova Server* locally. The filtered list of available cells is then forward to the API CELL where a cell is selected from this filtered list according to the original resource allocation process in OpenStack. Then, the requested virtual resource is created on a hardware resource selected by the cell locally. Consequently, the location filter does not affect the resource allocation in OpenStack but reduces the search space of the resource allocation algorithm to those hardware resources that are in compliance with the location parameters (i.e., hardware resources of cells that are located in countries with an *adequate level of protection*). In the case of allocating hardware resources for a virtual machine with back-up instance, the *location filter* applies the location policy for the virtual machine and its back-up instance individually. In particular, it is possible to schedule virtual machines and back-up instances on disjunct sets of cells, thereby enforcing their hosting on hardware resources of separated cells.

The implementation of **location-determined logging** consists of (i) the extension of the CEILOMETER AGENT COMPUTE by a plug-in that gathers the logging data of the hypervisors and (ii) extending the CEILOMETER COLLECTOR, CEILOMETER DATABASE, and CEILOMETER API to also process hypervisor logs [118]. The **hypervisor logs** are accessed via the VMI *libvirt* (cf. 'LibvirtAPI' in OpenStack documenation [157] and *libvirt* documenation [167]). By using the LIBVIRTAPI of OpenStack, logs are gathered independently of the used hypervisor, and therefore, logging is supported for a large number of hypervisors.[1] The **authenticity and integrity of logging** are addressed by using a public-private key signature scheme with an individual private key for each compute node for signing logging data and a corresponding public key for verifying the integrity and authenticity of the logging data [58].

For the purpose of **compliance reporting**, two new components were added to the *Horizon Server*: (i) the COMPLIANCE CHECKER evaluating the logging data of the *Ceilometer Server* and the *Nova Server* and (ii) the AUDIT BOARD visualising compliance reports on evaluation results [23]. Detected compliance breaches are wrong placements of virtual machines with respect to the location policy and location parameters and deviations in the virtual machine execution from the requested configuration (e.g., CPU, memory, and storage). For **logging evaluation**, the logging data provided by the CEILOMETER API is compared with the local logs of the NOVA SCHEDULER and the location policy stored at the *Nova Server*. The **compliance reports** are visualised via the AUDIT BOARD which is implemented as a plug-in of the DASHBOARD. There are three different views of the compliance reports: (1) the cloud customer's view, (2) the cloud provider's view, and (3) the view for authorised third parties. The views are implemented according to the monitoring and reporting architecture proposed in Section 5.4.3. Consequently, the cloud customer's view reports on virtual resources of the respective cloud customer only. The cloud provider's view reports on all hardware resources and virtual resources. The view for authorised third parties is implemented for two use cases with different information levels: (a) an investigation by a tax officer and (b) service operations controls by an external auditor instructed by a cloud customer. The information level pre-

[1] An overview of hypervisors supported by libvirt can be found in the Internet: http://libvirt.org/drivers.html (last visited: 30.06.2015).

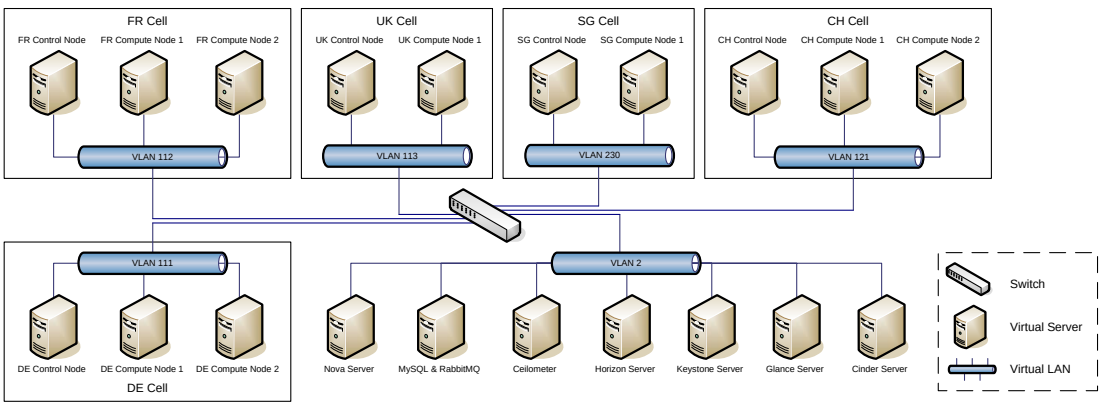
sented to a tax officer is limited to location information of virtual resources of the investigated cloud customer, while the external auditor has the same information level as the instructing cloud customer. Access to the different views is controlled via the access control mechanism of OpenStack. The cloud customer's view is accessible only by user accounts and the cloud provider's view only by admin accounts. For the views of authorised third parties, two additional roles are added to OpenStack: (a) *tax officer* and (b) *auditor*. Fo each role, access to the audit board is restricted to information appropriate to that role.

6.1.3 Experimental set-up and evaluation results

The experimental set-up is based on four physical servers which are virtualised using VMWARE ESXi in Version 5.1.0. All components of OpenStack are installed on virtual servers running Ubuntu 13.10. OpenStack is installed in release version 2013.2.3, 'Havana'. The only exception is the OpenStack Telemetry Ceilometer, which is installed in OpenStack release version 2014.1, 'Icehouse'. The version of the messaging bus RabbitMQ is 3.1.3.

Figure 6.2 illustrates **the experimental set-up** of virtual servers and virtual networks. In total, there are 20 virtual servers. Seven virtual servers are hosting the management components including OpenStack Compute Nova, OpenStack Dashboard Horizon, and OpenStack Telemetry Ceilometer, which is modified as described in Section 6.1.2. Further, there are 13 virtual servers organised in five OpenStack Compute Cells simulating a globally distributed cloud infrastructure. The simulated locations are Germany (DE Cell), France (FR Cell), Switzerland (CH Cell), the United Kingdom (UK Cell), and Singapore (SG Cell). Each cell is configured to operate in a dedicated VLAN and consists of a control node and either one or two compute nodes which is shown in Figure 6.2. The compute nodes use QEMU version 1.5.0 for the nested virtualisation of virtual machines. The nested virtualisation is controlled via libvirt using version 1.1.1. The CEILOMETER AGENT COMPUTE at the COMPUTE NODES is configured to log and forward the state of the hypervisor every ten minutes, which is the default setting of *OpenStack Telemetry Ceilometer*.

Figure 6.2: Experimental configuration of virtual servers and virtual networks.



In the experiment, the functionality of the location-determined resource allocation for

virtual machine placement as well as the logging and reporting mechanisms of the LDA are demonstrated. For that purpose, a set of eight location classes ${}^{loc}\mathbb{SC}$, which are illustrated in Figure 6.3, are defined according to Definition 5.42. Hardware resources of each cell are assigned the location class of the corresponding country, e.g., DE COMPUTE NODE 1 is assigned ${}^{loc}SC_{DE} \in {}^{loc}\mathbb{SC}$. Based on ${}^{loc}\mathbb{SC}$, a location policy is defined covering rules for two data types (i.e., personal data and financial data) and six origins (i.e., Germany, France, United Kingdom, Switzerland, Singapore, and EU/EEA). The rules of the location policy are illustrated in Figure 6.4. For each *data type* and *data origin*, a set of allowed cells $Cell_{user}$ is defined. Additionally, there is defined a set of cells $Cell_{bak}$ for hosting back-up instances.[1] When requesting a virtual machine with a dedicated back-up instance, the *location filter* of the NOVA SCHEDULER allocates the virtual machine at $Cell_{user} \backslash Cell_{bak}$ and the back-up instance at $Cell_{bak}$ and if $Cell_{user} = Cell_{bak}$ both instances are located at the same cell, i.e., $Cell_{user}$. For example, when requesting a virtual machine for processing personal data originating from Germany, the virtual machine is located at ${}^{loc}SC_{EU} \backslash {}^{loc}SC_{UK} = \{{}^{loc}SC_{DE}, {}^{loc}SC_{FR}, {}^{loc}SC_{UK}\} \backslash {}^{loc}SC_{UK} = \{{}^{loc}SC_{DE}, {}^{loc}SC_{FR}\}$ and the back-up instance is located at ${}^{loc}SC_{UK}$ (cf. Figure 6.4).

Figure 6.3: Location classes ${}^{loc}\mathbb{SC}$ in the experiment.

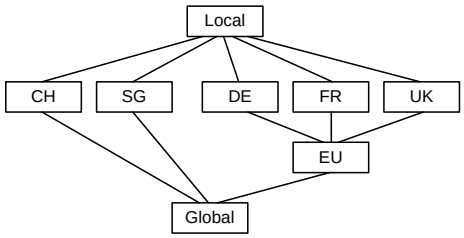


Figure 6.4: Location policy in the experiment.

Data origin	Data type: Personal data $Cell_{user}$	Data type: Personal data $Cell_{bak}$	Data type: Financial data $Cell_{user}$	Data type: Financial data $Cell_{bak}$
EU	EU	UK	EU	FR
DE	DE, FR	FR	DE	DE
FR	EU	DE	FR	FR
UK	EU	DE	UK	UK
CH	CH	CH	CH	CH
SG	SG	SG	SG	SG

The experiment consists of requesting four virtual machines with different configurations for *data type*, *data origin* and *back-up option*. All logging data of the experiment and sample screen shots of the modified DASHBOARD and AUDIT BOARD can be found in Appendix D. Table 6.1 lists the configuration of the virtual machines, the result of the decision by the location policy and the allocated compute nodes. First, three virtual machines configured to host financial data of different origins are requested, one after another. The rules on financial data are very restrictive (cf. Figure 6.4) and only allow the allocation of hardware resources that are located in the cell corresponding to the country of data origin. In the experiment, the compute nodes are selected accordingly to the decision by location policy. If the *back-up option* is selected the virtual machine and the back-up instance are distributed over both compute nodes of each selected cell. This implies that load balancing is active. Then, the forth virtual machine is configured to host personal data originating from the EU/EEA. In this case, the most relaxed rule applies since the location policy allows the allocation of any cell located within the EU/EEA. The dedicated cell for back-up instances is located in United Kingdom. The selected compute node for the virtual machine is DE COMPUTE 2 and for the back-up instance, UK COMPUTE 1. This is in compliance with the decision with respect to the location policy.

[1] $Cell_{user}$ and $Cell_{bak}$ are disjunct to demonstrate the concept of local separation in scenarios of disaster recovery.

Additionally, the resource allocation of the forth virtual machine illustrates the compliant operation of the *location filter* in several ways. First, compute nodes located within the EU/EEA are selected albeit there are unused compute nodes in the CH Cell (i.e., CH COMPUTE NODE 1) and SG Cell (i.e., SG COMPUTE NODE 1). This is in compliance with the location policy where CH Cell and SG Cell are not allowed for personal data originating from the EU/EEA. This implies that the decision with respect to the location policy outweighs load balancing. A second observation is that even within the cells located in the EU/EEA the virtual machine is placed on the already used compute node, DE COMPUTE 2, instead of being placed on the unused compute node, UK COMPUTE 1. Again, this is in compliance with the location policy according to which the UK Cell is reserved for back-up instances, and is therefore not available for the placement of the virtual machine itself. This implies that the decision by the location policy also outweighs load-balancing with respect to the disjointed placement of back-up instances. Finally, it is observed that the back-up instance is placed on compute node UK COMPUTE 1, which complies with the location policy. Screenshots of the DASHBOARD showing the extended virtual machine configuration, the overview on running compute nodes, and the overview on virtual machine instances can be found in Appendix D.3.

Table 6.1: Virtual machine configuration with resulting decisions and resource allocation

Virtual machine configuration				**Decision by location policy**		**Selected compute node**	
No.	*Data type*	*Data origin*	*Back-up*	$Cell_{user}$	$Cell_{bak}$	*Virtual machine*	*Back-up instance*
1	Financial data	Germany	yes	DE	DE	DE COMPUTE 2	DE COMPUTE 1
2	Financial data	Switzerland	no	CH	CH	CH COMPUTE 2	—
3	Financial data	France	yes	FR	FR	FR COMPUTE 2	FR COMPUTE 1
4	Personal data	EU/EEA	yes	EU	UK	DE COMPUTE 2	UK COMPUTE 1

Observations in the experiment are logging data of the hypervisors collected by the *Ceilometer Server* and logging data of the applied rules and filtered cells collected by the *Nova Server*. All logging data can be found in the Appendix D.1 and Appendix D.2. The structure and information content of the logs are exemplified by the request of the virtual machine no. 4.

Listing 6.1: *Nova Server* log-file of virtual machine no. 4.

```
1  [data]
2  type = "personal"
3  origin = "eu"
4  timestamp = "2014-11-27 12:32:15 UTC"
5  [accepted]
6  fr = true
7  de = true
8  sg = false
9  ch = false
10 uk = false
11 api = false
```

ID in OpenStack and file name: 1ADAFF73-6741-422B-B2F0-7C0E1A2459BD

Listing 6.2: *Nova Server* log-file of back-up instance of virtual machine no. 4.

```
1  [data]
2  type = "personal"
3  origin = "eu"
4  timestamp = "2014-11-27 12:32:16 UTC"
5  [accepted]
6  uk = true
7  sg = false
8  fr = false
9  ch = false
10 de = false
11 api = false
```

ID in OpenStack and file name: 6EDED12F-B120-4FC6-8C21-DAA4084630A0

Listing 6.1 and Listing 6.2 list the log-files recorded at the *Nova Server*. Each log-file is associated by file name with the virtual machine's ID in OpenStack. Further, each log-file contains the location parameter TYPE (i.e., data type) and ORIGIN (i.e., data origin) that the

Listing 6.3: Excerpt of ceilometer log for DE COMPUTE 2 before requesting virtual machine no. 4.

```
(
  ...
  [timestamp] => 2014-11-27T12:27:28
  ...
  [resource_metadata] => Array
  (
    ...
    [cell] => de
    [compute_node] => DeCompute2
    ...
    [instances.0] =>
  4b59295f-56a9-4126-8c07-4ff0e793f919
    [instances.amount] => 1
    ...
    )
  ...
)
```

Listing 6.4: Excerpt of ceilometer log for DE COMPUTE 2 after requesting virtual machine no. 4.

```
(
  ...
  [timestamp] => 2014-11-27T12:37:28
  ...
  [resource_metadata] => Array
    (
      ...
      [cell] => de
      [compute_node] => DeCompute2
      ...
      [instances.0] => 4b59295f-56a9
          -4126-8c07-4ff0e793f919
      [instances.1] => 1adaff73-6741
          -422b-b2f0-7c0e1a2459bd
      [instances.amount] => 2
      ...
    )
  ...
)
```

decision is based on and the timestamp of the decision. In addition, the result of the decision is listed for each available cell where TRUE indicates that the cell is accepted and FALSE that the cell was rejected. The logged decisions comply with the applicable rule of the location policy (cf. Figure 6.4). In addition, the timestamps imply that the decision on the placement of the virtual machine was made before that on the placement of the back-up instance. Listing 6.3 and Listing 6.4 show an expert ceilometer log for DE COMPUTE 2 before and after the virtual machine no. 4 is requested (cf. Listing D.18 and Listing D.19 for the complete log). In the listings, selected key-value-pairs of the JSON-object used by *OpenStack Telemetry Ceilometer* for representation are shown with respect to relevance for validating the placement of virtual machine no. 4. Each ceilometer logging entry particularly provides the timestamp of creation and information on hosting cell and compute node as well as the hosted instances by number and ID. The first excerpt is created before the request and indicates that there is only a single virtual machine running on DE COMPUTE 2. Then, after the request, the second excerpt of the log indicates that there is an additional virtual machine running which has the ID of the virtual machine no. 4 (cf. Listing 6.1). Consequently, virtual machine no. 4 is placed on DE COMPUTE 2 which is compliant with the observations made in the experiment (cf. Figure 6.1).

The cloud provider's view in the AUDIT BOARD visualises all active cells using an interactive world map where countries with running virtual machines are highlighted. Further, the AUDIT BOARD provides information on the virtual machines that are located in each cell and on the decisions made by LOCATION FILTER for each individual virtual machine. Screenshots of AUDIT BOARD can be found in the Appendix D.4.

In addition to the proof of concept, there are observations made on legal compliance, technical feasibility, and trustworthiness of the LDA which are investigated in Section 6.2.

6.2 The price and return on legally compliant cloud computing

The implemented LDA is not only a proof of concept showing that location determination is possible for cloud computing but also provides insight into the feasibility of legally compliant cloud computing in general and of location-determined cloud computing in particular. The LDA introduces legal compliance with respect to the location of data processing for the price of increased complexity in resource allocation and logging mechanisms. The same approach can be applied to confidentiality, integrity, and availability by using the corresponding security classes to classify cells and allowed information flow. The major differences are that the location policy is replaced by confidentiality, integrity, or availability policies and that the acquired logging data have to be extended by additional information on the respective security property to verify the resources' classification (cf. Section 5.4.1).

However, when implementing legally compliant cloud computing like the LDA, the question arises: At what price does legal compliance come and what is gained in return? Good indicators for the price are potential reductions in the functionality and usability of the cloud with respect to its characteristics (cf. Section 2.1.1). The gains are likely to be increased legal certainty and the increased trustworthiness of the technical implementation.

In the following, the impact on legal compliance and legal certainty are investigated in Section 6.2.1. Further, the observations on the feasibility of legally compliant cloud computing are examined in Section 6.2.2. Finally, the trustworthiness of the technical implementation is discussed in Section 6.2.3

6.2.1 Complying with legislation and corporate customers' requirements

In the scenario of IT outsourcing to the cloud, cloud providers have to comply with multiple legal requirements, which derive from the legal requirements of their corporate customers. The legal analysis in Chapter 3 identifies the general requirements that cloud providers have to implement technically (cf. Section 3.6). The identified requirements are: (1) the identification of the *necessary level of security*; (2) the implementation of security policies covering applicable legal and technical requirements; (3) the implementation and enforcement of safeguards covering (i) basic security measures, (ii) access control, (iii) transmission control, and (iv) countermeasures and incident response; and (4) compliance monitoring, documentation, and reporting. These technical requirements have to be implemented by the cloud providers in such a way that it is possible to support the legal requirements that apply to their corporate customers. To what extend the approach proposed in this thesis supports legal compliance is discussed in what follows.

The **identification of the *necessary level of security*** is based on the information on location, category of data, origin, and applicable requirements (cf. Section 3.6.1). The proposed approach is able to determine the location of hardware resources and support the placement of virtual resources according to existing location constraints (cf. Section 5.3.4). Further, it is possible to address additionally applicable requirements with respect to confidentiality, integrity, and availability (cf. Section 5.4.2). In the LDA proof-of-concept implementation, it is shown that providing the category of data and their origin is sufficient to request virtual machines which are then assigned to hardware resources in compliance with applicable location

constraints (cf. Section 6.1.2 and Section 6.1.3). Thus, cloud providers using the LDA are able to identify the *necessary level of security* and to assign virtual resources to hardware resources which are compliant with the *necessary level of security*. This includes the specific case of providing an *adequate level of protection* in terms of the European and German data protection law, since hardware resources are selected only if they are established at a location having an *adequate level of protection*. Moreover, it is possible to deal with location constraints generally, as is required in the case of German tax law (cf. Section 3.5.3.2) and export control (cf. Section 3.4.3).

This is achieved by implementing **security policies**, which describe the applicable rules for each corporate customer by assigning category of data and origin to a set of security classes corresponding to the *necessary level of security*. Due to the way information flow control methods are used, security classes directly translate into applied safeguards and security measures in respect of confidentiality, integrity, availability, and location constraints. Further, it is possible to extend the underlying information model used for information flow control and introduce additional security characteristics like non-repudiation and authenticity (cf. Section 7.3). However, new security characteristics increase the complexity of decision and enforcement in the cloud management process. Even though the complexity scales well with an increasing dimension of security characteristics, it still might not be feasible for very large numbers of security characteristics (cf. impact on rapid elasticity/scalability in Section 6.2.2). In any case, the presented approach fully supports the implementation of enforceable security policies expressing the individual legal requirements of each corporate customer. Moreover, corporate customers can specify their legal demands for each virtual machine individually by classifying them according to data category and origin. These specifications can be made in the outsourcing contract. Afterwards, corporate customers simply specify the data category and the origin on each virtual resource request, and the cloud provider can apply the rules of the security policies automatically when provisioning the requested resource.

The **implementation and enforcement of safeguards** at the cloud provider is supported by applying the rules of information flow control described in the security policies in an automated manner. Hardware resources are selected according to their security classification, which cover the implementation of basic security measures like confidentiality, integrity, and availability but also the location of their establishment. Further, it is possible to control the access to and transmission of virtual resources. This is done on the level of virtualisation management in the cloud management process, i.e., controlling the access and transmission with respect to hardware resources (i.e., controlling the *information flow of virtual resources*). The network access to virtual resources and resulting data transfer is in the administrative responsibility of the corporate customers, and is therefore not controlled by the cloud provider (cf. Section 5.1.2). The cloud provider has the administrative control of the *information flow of virtual resources*, and is therefore responsible for controlling it in compliance with law. This is possible for location constraints using the LDA (cf. Section 6.1.3) and can be extended to cover other security constraints (cf. Section 5.3.4). Further, it is possible to support countermeasures and incident response management, for instance, malicious/suspicious hardware resources can be isolated by changing their security classification in such a way that virtual resources are no longer assigned (cf. Section 7.3).

The **monitoring, documentation, and reporting of compliance** are particularly necessary when legal requirements on the inspection of the cloud provider apply, as they exist in German data protection law (cf. Section 3.2.2), German finance law (cf. Section 3.4.1), and German tax law (cf. Section 3.4.2). The inspection of the clouds can be done internally by the cloud provider, by the corporate customers (or an authorised third party) and by competent security authorities (cf. Section 3.2.2). Trustworthy monitoring, documentation and reporting in clouds is possible in general (cf. Section 5.4.3), and in particular, can be implemented in existing cloud infrastructures as it is shown for location constraints in the proof-of-concept implementation LDA (cf. Section 6.1.3). Moreover, specific inspection requirements for availability and locality can be supported. For example, for tax inspections of corporate customers, virtual resources can be assigned exclusively to hardware resources which are located in the territory where the competent tax office can exercise its inspection. Another example is when a search warrant has been issued for a corporate customer, the cloud provider can support the investigative authority by preventing the virtual resources of the searched corporate customer from being migrated to foreign countries (i.e., avoiding aiding the corporate customer by accidentally or intentionally preventing the search warrant).[1]

To conclude, the approach proposed in this thesis supports all technical requirements identified in the legal analysis (cf. Section 3.6). The cloud provider using this approach can place virtual resources in compliance with the legal requirements of their corporate customers. Further, each corporate customer can request virtual resources with specific legal requirements, and they are automatically provisioned by the cloud provider in compliance with legal requirements. Whether data processing in IaaS clouds is compliant or not depends on which legal requirements are specified by the corporate customers and on how consequently they are enforced by the cloud provider. Both are supported at the technical level by the approach proposed in this thesis.

6.2.2 Technical feasibility of legally compliant cloud computing

Based on the implementation work, the experiment, and general considerations, there are multiple observations made with respect to the impact of legally compliance cloud computing on cloud characteristics [141] [211], i.e., (1) resource pooling/virtualisation, (2) rapid elasticity/scalability, (3) on-demand self-service/pay-per use utility model, (4) measured service, and (5) broad network access (cf. Section 2.1.1), which are discussed in the following.

The **impact on resource pooling/virtualisation** is not considerable, since resource pooling and virtualisation are rather prerequisites than restrictions. When classifying hardware and virtual resources by location, it is still possible to organise resources by using virtualisation techniques and pooling them for provisioning. For legal compliance, it is sufficient that each resource pool can be classified correctly by location and *effective level of security*. For this purpose, resource allocation has to be made accordingly to the information flow allowed between the security class of the resource pool and security class of the assigned virtual resource. Usually, the clouds are organised in subsets of resource pools which are each operated by an

[1] This can be achieved by locking the virtual resources of searched corporate customer in place by changing their security classification to enforce the current placement, which can be the first step to initiate what is called a quick-freeze (cf. Section 3.5.2).

individual hosting site (cf. *global clusters* and *local clusters* in Section 4.1.4). Therefore, the security class of each resource pool is determined by the location of its operating hosting site and its *effective level of security*. Even if a resource pool is operated by multiple hosting sites, the security classes can be determined correctly by the least upper bound of the security classes of the involved hosting sites, i.e., by applying the $\otimes$-operator (cf. Section 5.3.1). Further, it is possible to cover differences in the *effective level of security* of individual hardware resources. In such a case, hardware resources of the same security class are pooled. Then, each resource pool consists of homogeneously classified hardware resources and is classified accordingly. In the LDA, resources are pooled by using *OpenStack Compute Cells* and virtualisation is performed by *compute nodes*. The management of cells is very flexible and compute nodes can be added and removed during run-time (cf. 'cells' in OpenStack documenation [157]). Since the LDA is based upon OpenStack Compute Cells, the resource pooling/virtualisation characteristic is a prerequisite for location determination and is not restricted by its implementation.

The **impact on rapid elasticity/scalability** can be considerable since security constraints can limit the hardware resources that are available for virtual resource allocation. Only hardware resources with sufficient security classes can be assigned if virtual resources require a specific location and the*ensured level of security*. If hardware resources with sufficient security classification are not available, additional virtual resources either cannot be created (*reject*) or have to be assigned to hardware resources with insufficient security classification (*SLA breach*). Alternatively, *reorganising* the embedding of virtual resources can help the offort to find a more efficient embedding, freeing hardware resources with sufficient security classification through reduction of over-provisioning (cf. evaluating embeddings in Section 5.4.2). Further, it is also possible to *degrade* the security classification of virtual machines to security classes where hardware resources are available. Which strategy is used (i.e., rejection, SLA breach, reorganisation, or degradation) usually depends on the contractual agreement between cloud provider and corporate customer. A common practice of cloud providers is to accept the SLA breach and refund the corporate customers for the degraded services (e.g., amazon's *EC2 service level agreements* on availability[1]). However, there are cases where service degradation and SLA breaches are not acceptable. For example, without explicit permission of the responsible tax office, it is forbidden to process data that are relevant to German tax law abroad (cf. Section 3.5.3.2). Here, a degradation or SLA breach can result in administration fees (e.g., BDSG §43, 44 and StGB §204) and custodial sentences (e.g., BDSG §44 and StGB §203, 204). Therefore, reorganisation and rejection are better strategies with respect to legal certainty than service degradation and accepted SLA breaches. In addition, it is possible to increase the number of hardware resources of the specific security classification to better fit the corporate customers' demands.

When applying metrics to evaluate the quality of the embedding (cf. Section 5.4.2), it is possible to determine the degree of over- and under-provisioning. This empowers the cloud provider to consider the legal and technical consequences when scheduling virtual resources. Therefore, rapid elasticity and scalability may be reduced when it is necessary to enforce le-

[1] Amazon charges their cloud customers for a 99.95% availability and refunds gradually when lower availability is provided; cf. *EC2 service level agreements* in the Internet: `http://aws.amazon.com/de/ec2/sla/` (Last visited: 30.06.2015).

gal compliance, due to a shortage of hardware resources with sufficient security classification. However, cloud computing remains rapid, elastic and scalable if hardware resources with sufficient security classification are available. The proof-of-concept implementation shows that the virtual machine configuration can be extended by security parameters, which allows automated decision-making and enforcement of resource allocation (cf. Section 6.1.3). The impact on run-time behaviour is not the focus of this thesis and, is therefore not investigated in the experiment. In any case, the impact can be estimated analytically. Each included security parameter (i.e., location, confidentiality, integrity, and availability) introduces an additional dimension to the NP-hard decision problem of virtual resource allocation (cf. Section 5.4.2). The complexity of existing algorithms solving this class of decision problems in general scales with the number of dimensions on a square-logarithmic base (vector scheduling), logarithmic base (vector bin packing), and linear base (knapsack problem) [39]. This is also observable for resource allocation algorithms for virtualised service hosting platforms and specifically cloud infrastructures [189]. Therefore, the impact of the additional dimensions on the run-time complexity of resource scheduling algorithms is rather low in comparison to the exponential complexity of finding optimal embedding. Investigations into the proof-of-concept implementation shows that also logging scales on a linear basis with respect to number of log entries generally and with respect to the number of logged compute nodes specifically [118].

The **impact on on-demand self-service and pay-per-use utility model** is not significant since corporate customers can order virtual resources on-demand with or without security constraints, which are then processed automatically by the cloud platform. In the proof-of-concept implementation, location constraints are specified by the corporate customer when requesting virtual resources in the DASHBOARD. The virtual resources are then scheduled and provisioned automatically, which facilitates the on-demand self-services. Further, the pay-per-use utility model can be directly applied on requested security constraints by charging based on provided security properties. For example, the cloud provider can charge an additional fee for requesting location-determined hosting to compensate the reduced flexibility in virtual resource placement due to location constraints. Therefore, introducing security constraints for legally compliant cloud computing fully complies with the characteristic of on-demand self-service and the pay-per-use utility model.

The **impact on measured services** is considerable since monitoring and reporting of legal compliance requires the measurement of security classifications and extended logging with respect to security properties (cf. Section 5.4.1 and Section 5.4.3). In the proof-of-concept implementation, existing logging mechanisms are used and extended to log security relevant information at the hypervisors, and the information is made available for monitoring and reporting in the DASHBOARD (cf. Section 6.1.2). Further, the implemented logging and monitoring architecture is fully automated (cf. Section 6.1.2) and scalable (cf. 'impact on on rapid elasticity/scalability' above). Therefore, the characteristic of measured services is feasible for legally compliant clouds which support scalable and automated service monitoring and reporting.

The **impact on broad network access** is in principle not significant since virtual resources remain accessible irrespective of the security properties that are applicable. This is because access to virtual resources has no impact on the *information flow of virtual resources* but on the *information flow of processed data* in virtual resources (cf. Section 5.1.2). However, the

control of access to virtual resources is relevant with respect to the location and the *effective level of security* at the point of access. For example, according to German data protection law, transmitting personal data to a third country in general is illegitimate if no explicit statutory permission applies, independent of whether these data are processed inside or outside of the cloud. Then again, the control of *information flow of processed data* – and therefore the access to virtual resources – is in the responsibility of the corporate customer (cf. Section 5.1.2). For that reason, there are no limitations in the broad network access from the cloud providers' perspective. Nevertheless, if the cloud provider is managing access control to virtual resources on behalf of the corporate customers then the cloud provider has to restrict the broad network access to communications end-points with respect to the end-points' location and their *effective level of security*. Consequently, there is an impact on broad network access with respect to control the *information flow of processed data* (which is usually handled by the corporate customers) but not with respect to control the *information flow of virtual resources* (which is handled by the cloud providers).

6.2.3 Trustworthiness of legally compliant cloud computing

For corporate customers, it is important that cloud providers fulfil the service contract and provide the virtual resources in compliance with legislation and SLAs. Therefore, cloud providers have to monitor the service delivery of hardware providers which operate the hardware resources, and cloud providers have to provide evidence of the delivery of legally compliant services by reporting to their corporate customers (cf. Section 3.6.4).

The monitoring and reporting architecture in cloud infrastructure proposed in Section 5.4.3 addresses this requirement by introducing hardware-based authenticity and integrity to logging of hardware resources and virtual resources. By these means, evidence of the operation of hardware resources by the hardware providers and on operation of virtual resources on hardware resources is gathered at the cloud provider's site. The logging provides information on hardware identifiers which allow the verification of location and security properties by manual inspection (cf. Section 5.4.1). Further, the logging provides evidence of the security classification of hardware resources and virtual resources which can be used to verify the resource allocation with respect to requirements on location and the *effective level of security* (cf. Section 5.4.2). In addition, the evidence of security classification is a basis for generating compliance reports on the delivery of virtual resource services to the corporate customers. The use of hardware identifiers in the reports helps to reference hardware resources when clarifying questions asked by the corporate customers or providing additional evidence on the hardware utilisation, for example, in the context of a lawsuit. Further, the corporate customers can assign an auditor to audit whether the service delivery is in compliance with law, for example in the context of the standards on service organisation controls SSAE 16 [7] and ISAE 3402 [107].

In the LDA proof-of-concept implementation, it is shown that it is possible to log at hypervisor level [118] and to ensure the authenticity and integrity of telemetry data [58]. In particular, the hardware resource's and virtual resources' identifier uniquely identify the resources which, in combination with the timestamps and information on assigned compute cells, allows the resource allocation and service delivery to be retrospectively verified. For example, in the excerpt of the ceilometer log of DE COMPUTE 2, after requesting virtual machine no. 4 in

the experiment (cf. Listing 6.4), the timestamp indicates the point in time when the log entry was created. Further, the virtual resource's identifiers of virtual machine no. 4 (and that of virtual machine no. 1) are contained in the log, and to identify hardware resource, the hardware resource's identifier and the corresponding public key are contained in the log, too (cf. Listing D.19 in Appendix D.2). The signature contained in the log allows the integrity of the log entry to be verified, and in combination with the public key, it is possible to perform authenticity checks on the hardware resource's identity by creating the log entry [58].

If one is not using officially regulated digital signature schemes (like in the German Signature Act and the European Directive 1999/93/EC on a Community framework for electronic signatures) the "suitability for providing evidence in legal proceedings is very low" [173]. Therefore, it is necessary to implement mechanisms for integrity and authenticity in compliance with applicable legislation. As a result, depending to the chosen signature scheme, the evidence of the signature may not be given in some countries. From a technical perspective, it is possible to use multiple signature schemes at the same time where each complies with specific applicable legislation, and collectively, all applicable legislations are covered. However, this increases the size of the logs, and therefore, requires additional computational resources to generate and verify the signatures. Further, from the perspective of economics, additional investments for each additionally used signature scheme is necessary (e.g., for buying signature key certificates at accredited certification authorities as required by the German digital signature act [173]). This implies that if the signature schemes used are not compliant with applicable legislation there is no legal certainty that cloud providers can provider evidence on service delivery, even if the signature schemes are considered best practices. Although the corporate customers may trust the evidence of a best practice signature scheme (like using X.509 security certificates of a Public Key Infrastructure (PKI), for example, the *Symantec Managed PKI Service* [194]) without a guarantee of legal certainty, because these best practices are usually widely accepted, de facto standards and their implementation often comply with IT-Security standards (e.g., *Symantec Managed PKI Service* [194] is certified to comply with SSAE 16 [7] / ISAE 3402 [107] and WebTrust [6]).

Consequently, the trustworthiness of compliance monitoring and reporting primarily depends on the integrity and authenticity of the logging. Technically, the integrity and authenticity of the logging data can be ensured through signature schemes. If the signature schemes used follow best practice, the trustworthiness can be considered high, but legal certainty is achieved only by signature schemes that comply with the applicable legislation.

6.3 Conclusions on legal and technical boundary

It is usually technically possible to support legal compliance of IT outsourcing to globally distributed IaaS clouds is generally possible. In particular, the cloud providers can aid their corporate customers in achieving legal compliance individually. Assuming that only trustworthy hardware resources are used in the cloud, the cloud provider can place virtual resources compliantly to the *necessary level of security* considering the *effective level of security* of the hardware resources. This includes requirements for *necessary legal framework conditions* at the hardware resource locations. This means the cloud provider is able to provide corporate

customers with cloud resources accordingly to their legal requirements. This empowers the corporate customers to request and utilise the cloud resources in compliance with legal requirements, which are applicable to their data processing in the cloud. In this way, the corporate customers keep control over their own compliance management, and can even verify the compliance of virtual resources used with the specified *necessary level of security* using the compliance monitoring and reporting architecture of the cloud. An analysis of the cloud characteristics shows that introducing legal compliance to the cloud management process has significant impact with respect to (i) increased storage complexity on monitoring and logging, (ii) increased computational complexity in decision on resource allocation and (iii) restricted flexibility of the hardware resource utilisation due to legal constraints. Nonetheless, the implementation of legal compliance is feasible and scales with increased complexity in computation and storage (linear complexity is an observable upper boundary in both cases).

This is a great step towards legally compliant IT outsourcing to clouds. However, there are limits that have to be respected when planing IT outsourcing. First of all, IT outsourcing is not admissible in all cases. In Germany for example, the outsourcing of accounts keeping requires explicit approval of the competent tax office (cf. Section 3.4.2), and hospital facilities and penal doctors must not outsource the billing of their patients (cf. Section 3.4.4). Further, cloud providers can support their corporate customers in their efforts to achieve legal compliance only according to the requirements specified by the corporate customers. It is up to the cloud customers to specify and utilise the cloud's resources in compliance with the legal requirements which are applicable to the processing of the data. Moreover, in the particular case of IT outsourcing to IaaS clouds, the cloud customers are responsible for all data processing done by applications running on virtual resources in clouds, since the corporate customer have administrative control over the applications. This may differ for PaaS clouds and SaaS clouds, where service providers have administrative control over the applications running in the cloud. Another limitation is that only legal requirements that refer to technical measures can be enforced in the cloud management process. In particular, organisational measures are not covered and have to be implemented separately – for instance, by defining operating instructions for the technical and administrative staff of the cloud providers and hardware providers. In addition, it is not possible to implement legal requirements that do not match with the technical characteristics of clouds. For instance, the regulation on direct inspection of the original IT system in German tax law prohibits remote enquiry, where the latter is de facto always the case in cloud computing (cf. Section 3.4.2). In such cases, exceptions have to be made to the regular data processing procedures, or explicit agreements with the competent authorities on legally compliant data processing have to be negotiated. For instance, the conflict with direct access of the original IT system in clouds, mentioned above, can be solved in two ways. The first way is to not outsource the original IT system to the cloud, which is very restrictive. The second way is to agree with the competent tax office that indirect access, which is also an option in the tax code, is sufficient for inspections (cf. Section 3.4.2).

To conclude, it is possible for cloud providers to support their corporate customers in achieving legal compliance. However, corporate customers remain responsible for controlling the legal compliance of their data processing in the cloud, which becomes possible when use the compliance monitoring and reporting architecture of the cloud.

Chapter 7

Conclusions and directions for future research

Starting with the two research questions, which legal requirements apply when outsourcing IT to clouds and how these requirements be can supported by cloud providers, this thesis contributes to the identification of the legal and technical boundary of cloud computing. Covering a legal and a technical analysis of data processing in clouds, the mutual dependencies of legislation and technology are investigated. One result of the understanding gained in this thesis is a proposal for an information-theoretical approach modelling the control of the *information flow of virtual resources* in clouds and its proof-of-concept implementation exemplifying the feasibility of location-determined data processing.

This chapter concludes the research work done in this thesis and provides an outlook on practical applications and directions of future research . In Section 7.1, the main contributions and results are reflected with respect to the initial research questions asked in the description of the problem (cf. Section 1.2) and with respect to the goal and objectives of the thesis (cf. Section 1.3). Section 7.2 discusses the application of the results and the impact on current cloud practices. An outlook on future research directions identified in this thesis is given in Section 7.3.

7.1 Main contributions and results

To answer the question on how to implement legally compliant cloud computing technically, the information-theoretical approach of this thesis is supported by a descriptive analysis of legal requirements in the European and particularly German legislation. In this analysis, the requirements which are applicable to cloud computing are identified. Using an interdisciplinary approach, this thesis identifies two synergies that can be exploited to achieve legally cloud computing. The first synergy is identified between different legal norms, which imply that there are technical requirements that generally apply in cloud computing. The second synergy is identified between the technical process of decision-making/enforcement and compliance monitoring and reporting. Here, the classification of cloud resources can be utilised to support both, and it forms the basis for achieving legally compliant cloud computing.

The technical requirements observed in the legal norms investigated indicate that there is a list of technical requirements which every cloud provider has to implement to support legally compliant IT outsourcing to clouds (cf. Section 3.6). In particular, ensuring the *necessary level of security* and the *necessary legal framework conditions* (including an *adequate level of protection* when processing personal data) in the cloud is key to achieving legal compliance. However, the technical analysis revealed that existing cloud platforms do not support this key requirement to the necessary extent. Research addresses these requirements only in the context of European data protection law and without having automated compliance monitoring and reporting in mind (cf. Section 4.2), ignoring the existence of synergies between different legal norms and between decision/enforcement and monitoring/reporting. However, the synergies identified can be exploited when using the classification of data categories defined in law, i.e., classification by location, category of data, origin, and applicable requirements. Then, it is possible to define an information model based on this classification which can be used for decision-making and enforcement as well as for compliance monitoring and reporting (cf. 'challenge of location inhomogeneity', Def. 4.6).

For that purpose, an information model is proposed in this thesis, which (1) classifies virtual resources and hardware resources and (2) describes the admissible assignment of virtual resources to hardware resources based on legal requirements of cloud customers. This is achieved by using a lattice-based model for the control of *information flow of virtual resources* in clouds which is implemented in the cloud management process (cf. Section 5.4). For the construction of the model, multiple steps are necessary. First, a cloud taxonomy and entity-relationship model is developed, describing virtual resources, hardware resources, and the cloud management process formally using sets and relations of an ontology (cf. Section 4.1). Second, an information flow analysis of clouds is performed, identifying that the *information flow of virtual resources* is key to controlling the information flow in clouds generally (cf. Section 5.1.2). Third, the existing methods of information flow control are extended to a general model on lattice-based information flow control (cf. Section 5.3.3), including the definition of the new types[1] of security classes for location and availability as well as the opportunity to introduce additional security classes on measurable[2] security characteristics. Fourth, the final step in adopting the general model to virtual resources and cloud computing (cf. Section 5.3.4 and Section 5.4).

The technical feasibility of the information model proposed in this thesis is shown in a proof of concept for location-determined data processing in clouds. For that purpose, the LDA is implemented based on the open-source Infrastructure-as-a-Service (IaaS) cloud platform, OpenStack, introducing location-determined resource management as well as compliance monitoring and reporting (cf. Section 6.1). Moreover, the technical requirements identified in the legal analysis can be implemented in cloud infrastructures generally (cf. Section 6.2.1). However, only technical aspects of the data processing can be considered by the approach proposed in this thesis. Organisational aspects like instructing personnel are not covered. They have to be implemented by the cloud provider additionally. Also, the *information flow of processed data*

[1] Existing security classes are know for their confidentiality [52] and integrity [175].

[2] Based on Denning's Axioms (cf. Theorem 5.2), the security classes have to be partially ordered and finite, and the existence of a lower bound and a least upper bound operator on the security classes is required.

whose control is in the responsibility of cloud customers is not addressed by the approach (cf. Remark 5.3). This is because control of this type of information flow can be decoupled from that of the *information flow of virtual resources* if virtual resources are utilised according to the security classification,[1] and therefore, it is not investigated in this thesis. An outlook on how cloud customers can control their information flows is given in Section 7.3. While decoupling the information flows in clouds enables the information flow control in cloud management process, it is also a technical boundary of legally compliant computing, since the responsibility for controlling the information flow has to be split. This is because the cloud provider cannot overcome the limitations of a services provider who has to follow the instructions of their client. Cloud providers cannot (and should not) control how their cloud customers utilise the cloud resources. However, this is a general restriction on IT outsourcing and not an exclusive restriction in cloud computing. Further, there is a legal boundary. The admissibility of IT outsourcing is regulated and so is outsourcing to clouds. IT outsourcing is not necessarily always admissible. The again, if IT outsourcing is admissible it must be possible to support the legal requirements of cloud customers. However, it is not possible to implement legal requirements that needs sorting technical characteristics of clouds. An example is the requirement for direct inspection of the original IT system under German tax law (cf. Section 3.4.2), which excludes the possibility of remote enquiries which are generally necessary in cloud computing. Even if there are not many such conflicts observable – only this one was identified in this thesis – they are possible and exist. In these cases, the cloud customers have to comply with the applicable regulations and if there is no legally compliant alternative cannot outsource to the cloud. The case of German tax law can be solved by agreeing with the competent tax office to accept remote enquiries since there exist alternative regulations for access to IT systems for the purpose of tax inspections (cf. Section 3.4.2).

In general, this thesis finds that the legal and technical boundary of cloud computing does not necessarily have to deviate from that of IT outsourcing generally. For legally compliant cloud computing the following items are important: (1) the transformation of the cloud customers' legal requirements into security policies that are enforced in the cloud management process and (2) the monitoring of the virtual resource assignment and its reporting to cloud customers. The approach presented in this thesis demonstrates that both are technically feasible.

7.2 Application and practical implications

The current practice in global cloud computing of assigning virtual resources is using a best-effort strategy, optimising the utilisation of hardware resources. However, this does not support the cloud customers the process their data in compliance with legal requirements. In particular, location-determined assignment of virtual resources is not guaranteed by cloud providers. There exist concepts on pooling resources by location of hardware resources, but commonly they are not used for supporting the legal compliance of the data processing. An exception from this are national clouds, which are operated by a cloud provider established in a single

[1] It is assumed in this thesis that this is possible in general (cf. *Assumption 9* in conj. with *Assumptions 4, 5, 6* in Section 5.1.3).

country and where only cloud resources located in that country are utilised. However, these national clouds do not fully support the flexibility of global clouds. Particularly in cases where data without location constraints needs reworking hosted in global clouds more cost-efficiently.

The approach presented in this thesis enables cloud providers of global clouds to support their cloud customers to achieve legal compliance while using the full flexibility of a global cloud infrastructure (within legal boundaries). Further, it is possible to implement national clouds in global clouds virtually by restricting the information flow of virtual resources to hardware resources located in a single country. Thereby, hybrid clouds consisting of global and national cloud computing are feasible. By introducing comprehensive monitoring of and reporting on the virtual resource assignment, it is further possible to empower the cloud customers to track the achieved legal compliance of the cloud resources utilised. However, there are still cases where national clouds cannot be combined with global clouds. This happens when there is need for dedicated cloud resources which are operated in a single country exclusively, for example the case of governmental cloud computing. In governmental cloud computing, cloud providers often have to specialise in the specific applications and requirements of their cloud customers (i.e., public administrations). The cloud providers have to be public administrators themselves when data are processed that are relevant to the responsibilities of public administrations, (cf. Section 3.4.5).

Another aspect is the processing of personal data in global clouds on behalf of European and particularly German companies. Currently, there is no guarantee that global clouds can support an *adequate level of protection* when processing personal data. To clear these doubts, cloud providers have to guarantee that personal data outsourced by European and German companies are processed only in countries that have an *adequate level of protection*. This is not the case for current market leaders in global cloud computing, and in addition many of them are established in the USA where the US Patriot Act applies, which conflicts with European data protection law. By controlling the *information flow of virtual resources*, these cloud providers are able to support the legal requirements of their customers technically and provide evidence on their enforcement. The conflict between US and European law cannot be solved through this approach, but it is technically possible to provide evidence on which legal norms have been applied by the cloud provider. In general, it is possible to assign a virtual resource containing personal data only to hardware resources in countries having an *adequate level of protection*. This also applies to any type of data that has to be processed within specific countries or on hardware resources with specific security requirements. In this way, cloud providers can help their cloud customers to achieve legal compliance. Further, the necessary trust in legally compliant cloud computing can be established by providing evidence of preformed data processing. This is done by using monitoring and reporting mechanisms which apply on the resource allocation process, i.e., cloud management process. Moreover, cloud customers can request virtual resources which have to comply with specific legal requirements on-demand. This perfectly supports the concept of self-service in cloud computing. As long as cloud resources which have the level of security requested (in respect to the applicable legal requirements) are available, the cloud resources can be provided automatically. However, the consideration of legal requirements for virtual resource assignment comes with the cost of limiting the flexibility of resource allocation and might result in reduced hardware utilisation and can therefore increase

the operating costs of virtual resources.

When looking into multi-cloud scenarios, it is important for cloud providers to connect with other cloud providers and, when it comes to achieving legal compliance, to negotiate the service levels (including the legal requirements) with each other. Here, it is important that applicable legal requirements can be communicated and their enforcement can be monitored and reported. The security classification of virtual and hardware resources provides a good basis for standardising the communication of these legal requirements and their enforcement, monitoring, and reporting. This also enables cloud providers to identify other cloud providers that have the capacity to support legal requirements – which supports the listing of cloud services in directory services – and to share cloud resources with them automatically and on demand, in compliance with the individual requirements of each cloud resource. Further, this enables cloud providers to report back to their cloud customers concerning cloud resource assignment and whether applicable legal requirements were properly considered. This unfolds a new dimension of supporting the achievement of legal compliance in multi-cloud scenarios, including the scenario of the "cloud of clouds" which – similar to the idea of "network of networks" of the Internet – interconnects cloud infrastructures globally and seamlessly.

7.3 Outlook on directions of future research

This thesis focuses on IaaS clouds, but cloud computing also covers the service models, **PaaS and SaaS**. The major difference between SaaS and IaaS is that cloud customers do not utilise virtual resources but rather applications provided to them on demand. For the data processed in these applications the same legal requirements may apply as to virtual resources in IaaS. The applications are provided by service providers and, therefore, they responsible for the *information flow of processed data* caused by running these applications. Further, the cloud customers have no control over how the applications are hosted (whereas in IaaS they do have). This is in the control of the service providers. For that reason, the service providers are responsible for controlling the *information flow of processed data* in the cloud. To address this, the model for information flow control has to be extended to the application level including classification of applications and modelling information flow caused by them. A starting point is the projection of applications to the virtual resources which are hosting them. Then, the control of *information flow of virtual resources* can be used for hardware resource allocation. Further, additional security requirements apply in SaaS [191], including assuring secure communication between virtual resources, implementing access control to applications, and dealing with the vulnerabilities of the applications. This results in new information flows and criteria for allowed information flows, which have to be modelled. Additionally, admissible information flows have to be considered for service orchestration, since only virtual resources which have proper security classification can be allocated to run the application. It is also necessary to identify and classify the connection endpoints of the applications to enable proper control of admissible information flows using network connections. This has to be considered when developing cloud services. Otherwise the application might connect to endpoints which do not have a sufficient security classification. There exist approaches for controlling information flow in the development of cloud services which can help to address this issue [13]. However, the

research in this area is in the wings. In the context of PaaS, additional requirements for operating the development platform apply. As with SaaS, it is possible to project the platform to the virtual resources utilised and apply the control of *information flow of virtual resources*. However, unlike with IaaS and SaaS there are two levels of SLAs, i.e., the SLAs with the service providers which use the cloud resources for developing the cloud applications, and the SLAs with the cloud customers which utilise the applications developed [81]. Basically, it is possible for the cloud provider to apply the SLAs of service providers and those of cloud customers individually, but the service providers also have to guarantee legally compliant operation of the developed cloud applications. Therefore, it is necessary that the service providers can control the information flow of virtual resources utilised for hosting the cloud applications as requested by the cloud customers. This can be achieved by classifying the applications individually to the cloud customers' demands and projecting these classification to the virtual resources utilised for hosting. Then it is possible to treat PaaS and SaaS similarly and implement information flow control on top of that used for IaaS.

The approach proposed in this thesis focuses on the control of *information flows of virtual resources*. However, there also exist **information flows of processed data and those of meta data** (cf. Section 5.1.1). These information flows also have to be controlled to achieve legal compliant data processing. Therefore, it is necessary to identify all possible information flows of each type and classify the involved subjects and objects. After having done this, it is possible to apply the general model on lattice-based information flow control described in Section 5.3.1, since it addresses the information flow between security classes generally. The identification and classification of the information flows of meta data can be complex, since these information flows can be individual for every type of hardware and software and usually involve multiple hardware and software providers. Generally, in this case, the meta data are considered objects while the software, hardware and the competent providers are considered subjects. Assuming that any data processed in virtual resources can become meta data, objects (i.e., meta data) are classified according to the classification of the virtual resource they origin from. Subjects are classified according to their need-to-know level and provided safeguards of the involved providers. This makes it possible not only to identify the allowed information flow but also to decide whether hardware or software are allowed to access the meta data of a specific virtual resource. However, this does not cover third-party recipients of meta data other than the competent providers (e.g., subcontractors, service organisations, and public authorities). In the model, these third-party recipients are considered subjects. In practice, they are difficult to identify without the cooperation of the relevant provider. Any third party that remains unidentified results in a covert channel undermining the control of the information flow of meta data. The *information flow of processed data* is generally in the individual control of every cloud customer (in the case of IaaS). As with processed data, information flows have to be identified and subjects and objects have to be classified. The data processed are considered objects, and the recipients accessing the data via connections to the virtual resource and the virtual resources themselves are considered subjects. The classification of the objects is done according to the legal requirements applying to them (similar to the classification of virtual resources). The classification of the subjects is done accordingly to their need-to-know level and the safeguards provided by them. Then, similar to meta data, it is possible to model

the information flow control using the general model on lattice-based information flow control described in Section 5.3.1.

A specific case of information flow control is the **access by authorised third parties**. This is because authorised third parties usually require access to a large number of objects, e.g., to all objects of a specific cloud customer in the case of an tax inspection. However, it is not necessary to provide access to those objects that are not relevant to the authorised third parties. For example, in the case of a tax inspection the competent tax authority must have access to tax data but not to personal data. Therefore, it is necessary to control the allowed information flows more specifically. Basically, it is possible to filter the access of the authorised third parties by security classifications, and provide access only to objects that have the security classification relevant to the authorised third party. In the example of the tax inspection, access is granted only to virtual resources that contain tax data. However, these virtual resources can also contain personal data, to which access then has to be accepted since access to tax data has priority. In any case, it is possible to control the access of authorised third parties in such a way that access is granted only to objects that contain data which are relevant to the authorised third party and that access to all other objects is excluded.

In this thesis, legal compliance is investigated for cloud infrastructures. There also are the **client systems accessing the cloud infrastructure**. To control the information flow from and to these clients, it is necessary to classify client systems as well as cloud resources. Classification of client systems can be complex since they can be located all over the world and are often mobile (e.g., in the scenarios of the mobile cloud and the internet of things [115]). In particular, client systems can change their classification when physically crossing a boarder into another country. It is possible to model the change of security classification, but this introduces additional complexity and may result in covert channels if not modelled properly [20]. Additionally, it is complex to verify the security classification remotely. In particular, the remote verification of the current location is limited to identifying the network access point, and physical inspection provides little or no evidence on the client system location in the past or the future. However, it is possible for client systems to provide data on their current location using location services such as Global Positioning System (GPS), which can be used to perform plausibility checks on the client system location.

The **support of countermeasures and incident response** is an important requirement for the secure operation of a cloud infrastructure (cf. Section 3.6.3.4). The approach presented in this thesis does not directly address this requirement since it is not the focus of this thesis. However, it is possible to support countermeasures and incident response by using the mechanisms of controlling the *information flow of virtual resources*. For example by isolating a malicious/-suspicious virtual resource for further inspection and preventing information flow with other virtual resources (or client systems). It is also possible to consider hardware resources that are attacked or operating in a degraded mode. Both are possible by either changing the security class of the degraded security characteristic or flagging the resource with a new one. This allows an effective response to incidents that have an effect on the security characteristics of cloud resources, including the use of isolation as a countermeasure against malicious behaviour of cloud resources.

In general, it is possible to **introduce new security characteristics** to the information

model presented in this thesis (cf. Remark 5.7). Possible candidates for introduction are the authenticity of objects and subject (i.e., the truthfulness of their identity, origin, and provided safeguards) and non-repudiation (i.e., an action cannot be repudiated due to proof of its existence, e.g., due to audit-proof logging). Based on Denning's Axioms (cf. Theorem 5.2), new security characteristics have to be modelled by security classes which are partially ordered and finite, and where the existence of a lower bound and a least upper bound operator is given. Both authenticity and non-repudiation can be modelled by a set of two security classes using the class 'high' if the characteristic is guaranteed and 'low' otherwise [174].

In principle, it is possible to also introduce security characteristics addressing the **requirements of property-rights-protected data**. Property rights basically address origin/owner of the data and the permitted actions allowed on the protected data. This is very similar to controlling the access to processed data and the information flow control of processed data. However, there might be differences which have to be identified. Therefore, it is necessary to investigate the legal requirements that generally and specifically apply in property right law and derive applicable technical requirements as well as the security characteristics that have to be enforced. Possible candidates for security characteristics that can be modelled by the approach presented in this thesis are confidentiality (if data are not allowed to be shared freely), integrity (if data are not allowed to be modified freely), and authenticity (to ensure the origin of data and the identity of owners and accessing subjects).

Another topic of interest for further investigation is the **verification of security policies**. The security polices used in the experiment are manually derived from the information model and not very complex (i.e., there is only a small number of rules) due to the investigation of a small number of hardware resources and a single cloud customer. The complexity of the security policies increases with the increasing number of hardware resources, security classes and cloud customers. With increased complexity, the likelihood of introducing errors and interference between different security policies increases, too. Here, verification of security policies is necessary. In the thesis, this issue is considered by defining an XSD describing the structure of the security policies. More advanced verification is necessary in multi-cloud scenarios where security policies are exchanged between cloud providers [88]. Whenever multiple policies exist there is the risk that these policies will interfere, i.e., that there will be a conflict between the rules of different policies. Therefore, it is necessary to investigate how approaches to non-interference policies [193] can be used in the cloud management process and for exchanging security polices with other cloud providers.

This thesis takes a first step towards an **IaaS cloud computing ontology** by proposing a taxonomy and an entity-relationship model for virtual resources, hardware resources and the cloud management process (cf. Section 4.1). The next step is to perform a formal concept analysis to verify the consistency and completeness of the model. This can be done by using a formal concept analysis tool such as "The Concept Explorer"[1] and the "ToscanaJ Suite".[2]

[1]Documentation of "The Concept Explorer" on the Internet http://conexp.sourceforge.net/ (last visited: 30.06.2015).

[2]Documentation of the "ToscanaJ Suite" on the Internet http://toscanaj.sourceforge.net/ (last visited: 30.06.2015).

Appendix A

Comparison of virtual resources in current cloud infrastructres

Three commercial and two open cloud infrastructures are selected as reference for existing cloud infrastructures and are analysed on the basis of current documentation and literature. For the commercial cloud infrastructures, the following candidates are selected:

- Amazon Web Service (AWS) [5] is selected for its mature concept of distributed resource provision. Additionally, it is well documented in literature.
- Windows Azure [143] is selected as one of the major competitors of AWS, which unlike from AWS provides both PaaS and IaaS. Windows Azure is well documented as well.
- Fujitsu Cloud IaaS Trusted Public S5 is selected because of its mature concepts for high availability, which are beyond current standards in practice.

Following the recommendation of Voras et al., OpenStack [157] and OpenNebula [156]) are selected as candidates for open cloud infrastructures.

Based on the investigation of these cloud infrastructures, a classification of IaaS computing resources is provided in Section 4.1.2.

In the following, the summary of the comparison is listed in four tables, one for virtual machines (cf. Tab. A.1), one for virtual storages (cf. Tab. A.2), one for virtual links (cf. Tab. A.3), and one for virtual network (cf. Tab. A.4).

Table A.1: Survey on virtual machine properties

Property	Amazon Web Service (AWS)[5]	Windows Azure[143]	Fujitsu Cloud IaaS Trusted Public S5[126][75]	OpenStack[157]	OpenNebula[156]
Regular	• Standard: m3, m1	• Standard: A0,A1,A2,A3,A4	• Standard: Econonmy, Standard, Advanced, High-Performance, Double-High-Performance, Quard-High-Performance	• Standard: customisable 'Flavor'	• Standard: customisable virtual machine models
Special HW Requirements	• Data processing (incl. hyper-threading): c3, c1, cc2 • GPU (incl. HW-based en-/decoding): g2, cg1 • RAM: m2, cr1 • Storage (incl. high I/O performance): i2, hs1, hi1	• RAM: A5,A6,A7	• Private cloud: Dedicated server	• Customised requirements (by resource pooling): OpenStack Compute Cell, availability zones	• Customised requirements (by resource pooling): Clusters, oZones, Virtual Data Centers
Specialised Image	• Customized image: Amazon Machine Image (AMI) • Office environment: Amazon WorkSpaces	• Customised image: Virtual Hard Disk (VHD) • Infrastructure for PaaS: Web Sites (web server), Windows Azure Mobile Services (mobile application hosting), Windows Azure Cloud Services (cloud service hosting)	• Pre-defined images: Windows Server, Internet Information Server, CentOS, RedHat Linux RHEL • Middleware service: Microsoft SQL Server	• Customised image: several supported image formates (e.g. AMI, VHD, ISO)	• Customised image: several supported image formates (e.g. AMI, ISO)
Special Service Requirements	• Provisioning: reserved instances, on-demand instances, spot instances (bidding) • Utilisation (only reserved instances): light, medium, heavy • Scaling: auto scaling • Resource pooling: Clusters for data processing (c3,c1,cc2)	• None	• High availability: default	• Resource pooling: OpenStack Compute Cell, availability zones • High availability: OpenStack High Availability	• High avialability: OpenNebula High Avialability
Fault Tolerance	• Backup/Replication/Recovery: cp. virtual storage (only image and storage; RAM and SWAP is lost)	• Backup/Replication: cp. virtual storage (only storage; RAM is lost) • Recovery: Windows Azure Recovery Manager (RAM is lost)	• Backup/Replication: cp. virtual storage (only storage; RAM is lost) • Redundancy/Recovery: chassis-internal mirroring (RAM is lost)	• Backup/Replication/Recovery: snapshot (only volume storage; ephemeral storage and RAM is lost) in conj. w. OpenStack Block Storage	• Backup/Replication:: snapshot (of the VM's state) • Recovery: OpenNebula High Avialability

Table A.2: Survey on virtual storage properties

Property	Amazon Web Service (AWS)[5]	Windows Azure[143]	Fujitsu Cloud IaaS Trusted Public S5[126][75]	OpenStack[157]	OpenNebula[156]
Durability	• Non-persistent: Amazon EC2 Instance Store • Persistent: Amazon EBS, Amazon S3	• Non-presistent: Windows Azure Storage: Temporary Disk, Windows Azure Cache • Persistent: Windows Azure Storage (OS disk, data disk), Windows Azure SQL Database, Windows Azure HDInsight, Windows Azure Backup	• Non-persistent: System Disk • Persistent: Storage Disk	• Non-persistent: OpenStack Block Storage (Ephemeral Storage, Swap Disk, Root Disk) • Persistent: OpenStack Block Storage (Volume Storage), OpenStack Object Storage	• Configurable: non-/persistent
Connection	• Local: Amazon EC2 Instance Store • Remote: Amazon EBS, Amazon S3	• Remote: Default	• Remote: Default	• Local: OpenStack Block Storage, OpenStack Object Storage • Remote: OpenStack Block Storage (Ephemeral Storage, Volume Storage)	• Configurable: local/remote
Architecture	• Block storage: Amazon EC2 Instance Store, Amazon EBS • Object storage: Amazon S3	• Block storage: HDInsight • Object storage Windows Azure Storage, Windows Azure SQL Database, Windows Azure Cache, Windows Azure Backup	• Block storage: Default	• Block storage: OpenStack Block Storage (Ephemeral Storage, Volume Storage) • Object storage: OpenStack Object Storage	• Block storage: System Storage, Image Storage • Object storage: File Storage
Special Service Requirements	• Rare access: Amazon Glacier • Data transfer: AWS Import/Export • Connecting on-premises: AWS Storage Gateway • Content distribution: Amazon Cloud Front	• Caching + high I/O: Windows Azure Cache, Windows Azure Storage (optional disk chaching) • Content distribution: Windows Azure Storage (optional Disk)	• Detached storage archive: Disk Stand-by Area • Encryption: Storage Disk • Data erasure: Storage Disk • High availability: default	• Content distribution: OpenStack cluster • Storage migration: OpenStack Block Storage (optional)	• Content distribution: optional
Special HW Requirements	• SSD-support: in conj. w. storage optimised VMs (e.g. i2,hi1)	• None	• Private cloud: Dedicated server	• None (planed: resource pooling)	• Customised requirements (by resource pooling): Clusters, oZones, Virtual Data Centers
Fault Tolerance	• Backup/replication/recovery: snapshot in conj. w. Amazon S3 (distributed in conj. w. availability zones)	• Distributed replication: Windows Azure Storage(local- and geo-replication) • Backup: Windows Azure Backup • Recovery: Windows Azure Recovery Manager	• Redundancy: mirroring of storage chassis (default) • Backup/Restore: System Disk, Storage Disk • Distributed backup: Fujitsu Backup as a Service	• Backup/replication/recovery: OpenStack Block Storage (snapshot) • Distributed Backup/replication/recovery: OpenStack Object Storage	• Backup/Recovery: automatically on VM's shut-down and start • Replication: clone storage

Table A.3: Survey on virtual link properties

Property	Amazon Web Service (AWS)[5]	Windows Azure[143]	Fujitsu Cloud IaaS Trusted Public S5[126][75]	OpenStack[157]	OpenNebula[156]
Access network	• Private: AWS Direct Connect • Public: Default	• Private: ExpressRoute, Windows Azure Virtual Network • Public: Windows Azure Web Sites / Mobile Services / Cloud Services	• Private: VPN Connection Service • Public: Internet Connection Service	• Private: OpenStack VPNaaS • Public: Default	• Private: Default • Public: Optional
Special HW Requirements	• Specialised access router: AWS Direct Connect	• Specialised access router: ExpressRoute	• None	• None	• None
QoS Requirements	• Short delay + high performance: Clustering, Enhanced Networking	• High availability: Windows Azure Virtual Network/Mobile Services/Cloud Services	• High availability: Default	• High availability: OpenStack High Availability	• None
Fault Tolerance	• Unknown	• Redundancy (2oo2): ExpressRoute	• Redundancy: network devices and LAN cabling (default)	• None	• None

Table A.4: Survey on virtual network service properties

Property	Amazon Web Service (AWS)[5]	Windows Azure[143]	Fujitsu Cloud IaaS Trusted Public S5[126][75]	OpenStack[157]	OpenNebula[156]
VPN Management	• Virtual Private Cloud	• Windows Azure Virtual Network	• Default	• OpenStack VPNaaS	• Default
Load Balancing	• Elastic Load Balancing	• Windows Azure Traffic Manager	• Server Load-balancing Service	• OpenStack Object Storage • OpenStack High Availability • OpenStack LbaaS	• in conj. w. Clusters
DNS	• Amazon Route 53	• Windows Azure Traffic Manager	• Firewall Service	• Default	• Default
Proxy/Cache	• Elastic Load Balancing	• None	• None	• OpenStack Object Storage • OpenStack High Availability	• None
Gateway / Firewall	• Virtual Private Cloud	• Windows Azure Web Sites • Windows Azure Mobile Services • Windows Azure Cloud Services	• Firewall Service	• OpenStack FwaaS • OpenStack Layer-3 Networking	• Default
DHCP	• Default	• Default	• Default	• Default	• Default
Fault Tolerance	• Unknown	• Unknown	• Unknown	• Unknown	• Unknown

Appendix B

Construction of a lattice-based system using confidentiality classes

In the first step of the construction, classifications in $\mathbb{C}$ and categories in $\mathbb{K}$ are mapped to security classes in $\mathbb{SC}$, which are denoted *confidentiality classes*. The allowed information flow between confidentiality classes is defined on bases of the comparison of classifications and categories which are mapped to the respective confidentiality classes. Then, confidentiality classes are defined according to Definition 5.23.

In the second step of construction, the classification/needs-to-know vectors are replaced by security bindings. Therefore, it is necessary to redefine all parts of the model that are based on classification/needs-to-know vectors which are *state*, *request*, *request sequence*, and *system*. As a naming convention and to allow distinguishing between the original and the redefined terms, the newly introduced terms will carry the prefix *'c-'* where *'c'* indicates the context of *confidentiality*. For example the redefine *state* will become *c-state*.

State is redefined by replacing the classification/needs-to-know vector with security bindings of confidentiality classes. Consequently, *state sequence* has also to be redefined to apply to the new definition of state.

Definition B.1 (C-state) *A c-state ${}^cV$ is defined (analogously to state; cf. Def. 5.10) as a triple ${}^cV := (b, M, {}^cSCB)$ with*

- *$b \subseteq \mathbb{S} \times \mathbb{O} \times \mathbb{A}$ set of all subjects $S \in \mathbb{S}$ having access to objects $O \in \mathbb{O}$ in what access mode, which is described by a set of access attributes $A \subseteq \mathbb{A}$;*
- *$M \in \mathbb{M}$ access matrix in the state ${}^cV$; and*
- *${}^cSCB \subseteq \mathbb{SCB}$ set of confidentiality security bindings describing the binding of confidentiality classes to subjects and objects, and ${}^c\mathbb{SCB} \subseteq \mathbb{SCB}$ set of all confidentiality security bindings in $\mathbb{SCB}$.*

Then, ${}^c\mathbb{V}$ is the set of c-states ${}^cV_i$.

Definition B.2 (C-state sequence) *A c-state sequence is (analogous to state sequence; cf. Def. 5.11) an arbitrary number of chronologically ordered c-states ${}^{c}V_i \in {}^{c}\mathbb{V}$. Then, ${}^{c}\mathbb{Z} : {}^{c}\mathbb{V}^{\mathbb{N}}$ is the set of request sequences ${}^{c}Z_i$.*

Further, *request* has to cover that subjects may change security bindings of objects. Therefore, it is necessary to replace the set of classification/needs-to-know vectors by the set of security bindings of confidentiality classes. Consequently, *request sequence* has also to be redefined to apply to the new definition of *request*.

Definition B.3 (C-request) *A c-request is defined (analogously to request; cf. Def. 5.12) as a quadruple $(S_1, S_2, O_s, {}^{c}G) \subset \mathbb{S}^{+} \times \mathbb{S}^{+} \times \mathbb{O} \times {}^{c}\mathbb{G}$ with ${}^{c}\mathbb{G} := \mathbb{A} \cup \emptyset \cup {}^{c}\mathbb{SCB}$. Then, ${}^{c}\mathbb{R} : \mathbb{S}^{+} \times \mathbb{S}^{+} \times \mathbb{O} \times {}^{c}\mathbb{G}$ is the set of requests ${}^{c}R_i$.*

Definition B.4 (C-request sequence) *A c-request sequence is defined (analogously to request sequence; cf. Def. 5.13) as an arbitrary number of chronologically ordered c-requests ${}^{c}R_i \in {}^{c}\mathbb{R}$. Then, ${}^{c}\mathbb{X} : {}^{c}\mathbb{R}^{\mathbb{N}}$ is the set of c-request sequences ${}^{c}X_i$.*

Then, *system* is redefined by applying the new definitions of state and request as follows.

Definition B.5 (C-system) *Let ${}^{c}\overline{W} \subset {}^{c}\mathbb{R} \times \mathbb{D} \times {}^{c}\mathbb{V} \times {}^{c}\mathbb{V}$. A c-system ${}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0) \subset {}^{c}\mathbb{X} \times \mathbb{Y} \times {}^{c}\mathbb{Z}$ is defined (analogously to system; cf. Def. 5.17) by $({}^{c}X, Y, {}^{c}Z) \in {}^{c}\Sigma({}^{c}\mathbb{R}, \mathbb{D}, {}^{c}\overline{W}, {}^{c}z_0)$ if and only if $({}^{c}X_t, Y_t, {}^{c}Z_t, {}^{c}Z_{t-1}) \in {}^{c}\overline{W}$ for each $t \in \mathbb{N}$ where ${}^{c}z_0 := (\emptyset, M, {}^{c}SCB)$ is initial state with $M \in \mathbb{M}$ initial access matrix and ${}^{c}SCB \subseteq {}^{c}\mathbb{SCB}$ initial security bindings. ${}^{c}\overline{W}$ is considered* c-state transition relation.

Based on the *simple-confidentiality property* (cf. Def. 5.24) and the *confidentiality $*$-property* (cf. Def.5.25=, the rules for a secure system are redefined.

Definition B.6 (C-rule) *A c-rule is (analogously to a rule; cf. Def. 5.20) a function ${}^{c}\rho : {}^{c}\mathbb{R} \times {}^{c}\mathbb{V} \to \mathbb{D} \times {}^{c}\mathbb{V}$. A c-rule maps a c-request and a c-state to a decision and a c-state.*

A c-rule ${}^{c}\rho$ is security persevering *if and only if*

$$\forall({}^{c}R, {}^{c}V) \in {}^{c}\mathbb{R} \times {}^{c}\mathbb{V}\ \exists D \in \mathbb{D}\ \exists\, {}^{c}V' \in {}^{c}\mathbb{V} :$$
$${}^{c}\rho({}^{c}R, {}^{c}V) = (D, {}^{c}V') \wedge {}^{c}V \text{ is secure c-state} \Rightarrow {}^{c}V' \text{ is secure c-state.}$$

Analogously, a c-rule ${}^{c}\rho$ is $*$-property preserving *if and only if c-state ${}^{c}V$ satisfies confidentiality $*$-property implies c-state ${}^{c}V'$ satisfies confidentiality $*$-property.*

The handling of c-requests by a c-rule and the response of a c-system are defined analogously to rules and systems (cf. Def. 5.20).

Definition B.7 (10 c-rules for a secure c-system) *Analogously to the rules of a secure system Ω defined by LaPadula et al. [127], ${}^{c}\Omega := \{{}^{c}\rho_1, ..., {}^{c}\rho_{10}\}$ is the set of c-rules of a secure c-system where ${}^{c}\rho_i := \rho_i$ for $i \in \{3,5,6,7,9,10\}$ and ${}^{c}\rho_1, {}^{c}\rho_2, {}^{c}\rho_4, {}^{c}\rho_8$ are defined with ${}^{c}\overline{scb}(S), {}^{c}\overline{scb}(O), {}^{c}\overline{scb}(O') \in {}^{c}\mathbb{SCB}$ corresponding security bindings of subject $S \in \mathbb{S}$ and objects $O, O' \in \mathbb{O}$:*

- ***C-Rule 1 (get-read)** ${}^{c}\rho_1$**:** A subject S gets read access to an object O if*

(i) (security preserving) the access attribute $\underline{r}$ is element of the corresponding entry of the access matrix, and ${}^{c}\overline{scb}(O) \mapsto {}^{c}\overline{scb}(S)$; and

(ii) ($*$-property preserving) for all objects O' where S can write to (i.e., access in append and write mode) is true: ${}^{c}\overline{scb}(O) \mapsto {}^{c}\overline{scb}(O')$.

- ***C-Rule 2 (get-append)*** ${}^{c}\rho_2$***:*** *A subject S gets append access to an object O if*

  (i) (security preserving) the access attribute $\underline{a}$ is element of the corresponding entry of the access matrix; and

  (ii) ($*$-property preserving) for all objects O' where S can read from (i.e., access in read and write mode) is true: ${}^{c}\overline{scb}(O') \mapsto {}^{c}\overline{scb}(O)$.

- ***C-Rule 4 (get-write)*** ${}^{c}\rho_4$***:*** *A subject S gets execute access to an object O if*

  (i) (security preserving) the access attribute $\underline{w}$ is element of the corresponding entry of the access matrix, and ${}^{c}\overline{scb}(O) \mapsto {}^{c}\overline{scb}(S)$; and

  (ii) ($*$-property preserving [append]) for all objects O' where S has append access is true: ${}^{c}\overline{scb}(O) \mapsto {}^{c}\overline{scb}(O')$.

  (iii) ($*$-property preserving [read]) for all objects O' where S has read access is true: ${}^{c}\overline{scb}(O') \mapsto {}^{c}\overline{scb}(O)$.

  (iv) ($*$-property preserving [write]) for all objects O' where S has write access is true: ${}^{c}\overline{scb}(O') = {}^{c}\overline{scb}(O)$.

- ***C-Rule 8 (change-*** ^{c}***SCB)*** ${}^{c}\rho_8$***:*** *A subject S can change the security bindings ${}^{c}SCB \subseteq {}^{c}\mathbb{SCB}$ if*

  (i) (security preserving/$*$-property preserving) S changes only security bindings of objects where no subject has access to.

- ***C-Rules 3, 5, 6, 7, 9, 10*** *are constructed analogously to the rules 3, 5, 6, 7, 9, 10 (respectively) of Ω, since these rules describe a general system behaviour, which does not change when modelling confidentiality.*

With conferring the definition of allowed information flow between confidentiality classes (cf. Def. 5.23), it is easy to see that the c-rules ${}^{c}\rho_1$, ${}^{c}\rho_2$, and ${}^{c}\rho_4$ are defined analogously to the corresponding original rules ρ_1, ρ_2, and ρ_4. Further, c-rule ${}^{c}\rho_8$ is corresponding to rule ρ_8 where the classification/needs-to-know vector is replaced by the security bindings. In all other rules, the decision does not depend on the classification of subjects and objects, and therefore no modification is required.

Appendix C

XML-based location policies

To assign virtual machines to *nova compute cells* in OpenStack in the experiment, XML-based security policies are defined, which are used in the NOVA SCHEDULER (cf. Section 6.1.3). In the following the XSD and the security policies used in the experiments are given [19].

C.1 XML Schema Definition

Listing C.1: XML Schema definition for security policies used in the experiment.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="decision">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="datatype" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="origin" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="cell" maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="xs:boolean">
                          <xs:attribute name="name" type="xs:string" use="required"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                <xs:attribute name="name" type="xs:string" use="required"/>
                <xs:attribute name="default" type="xs:boolean" use="required"/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="name" type="xs:string" use="required"/>
          <xs:attribute name="default" type="xs:boolean" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="default" type="xs:boolean" use="required"/>
  </xs:complexType>
</xs:element>

</xs:schema>
```

C.2 Example policies used in the experiment

Listing C.2: Policy to allocate virtual machines without backup instance.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<decision xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="celldecisionmatrix.xsd"
          default="true">
  <datatype name="personal" default="false">
    <origin name="global" default="true" />
    <origin name="eu" default="false">
      <cell name="de">true</cell>
      <cell name="fr">true</cell>
      <cell name="uk">true</cell>
      <cell name="ch">false</cell>
    </origin>
    <origin name="de" default="false">
      <cell name="de">true</cell>
      <cell name="fr">false</cell>
      <cell name="uk">false</cell>
      <cell name="ch">false</cell>
    </origin>
    <origin name="fr" default="false">
      <cell name="de">true</cell>
      <cell name="fr">true</cell>
      <cell name="uk">true</cell>
      <cell name="ch">false</cell>
    </origin>
    <origin name="uk" default="false">
      <cell name="de">true</cell>
      <cell name="fr">true</cell>
      <cell name="uk">true</cell>
      <cell name="ch">false</cell>
    </origin>
    <origin name="ch" default="false">
      <cell name="de">false</cell>
      <cell name="fr">false</cell>
      <cell name="uk">false</cell>
      <cell name="ch">true</cell>
    </origin>
    <origin name="sg" default="false">
      <cell name="sg">true</cell>
    </origin>
  </datatype>
  <datatype name="financial" default="false">
    <origin name="global" default="true" />
    <origin name="eu" default="false">
      <cell name="de">true</cell>
      <cell name="fr">true</cell>
      <cell name="uk">true</cell>
    </origin>
    <origin name="de" default="false">
      <cell name="de">true</cell>
    </origin>
    <origin name="fr" default="false">
      <cell name="fr">true</cell>
    </origin>
    <origin name="uk" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="ch" default="false">
      <cell name="ch">true</cell>
    </origin>
    <origin name="sg" default="false">
      <cell name="sg">true</cell>
    </origin>
  </datatype>
</decision>
```

Listing C.3: Policy to allocate virtual machines with backup instance.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<decision xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="celldecisionmatrix.xsd"
          default="true">
  <datatype name="personal" default="false">
    <origin name="global" default="true">
      <cell name="uk">false</cell>
      <cell name="sg">false</cell>
    </origin>
    <origin name="eu" default="false">
      <cell name="de">true</cell>
      <cell name="fr">true</cell>
      <cell name="uk">false</cell>
    </origin>
    <origin name="de" default="false">
      <cell name="de">true</cell>
      <cell name="fr">true</cell>
    </origin>
    <origin name="fr" default="false">
      <cell name="de">false</cell>
      <cell name="fr">true</cell>
      <cell name="uk">true</cell>
    </origin>
    <origin name="uk" default="false">
      <cell name="de">false</cell>
      <cell name="fr">true</cell>
      <cell name="uk">true</cell>
    </origin>
    <origin name="ch" default="false">
      <cell name="ch">true</cell>
    </origin>
    <origin name="sg" default="false">
      <cell name="sg">true</cell>
    </origin>
  </datatype>
  <datatype name="financial" default="false">
    <origin name="global" default="true">
      <cell name="uk">false</cell>
      <cell name="sg">false</cell>
    </origin>
    <origin name="eu" default="false">
      <cell name="de">true</cell>
      <cell name="fr">false</cell>
      <cell name="uk">true</cell>
    </origin>
    <origin name="de" default="false">
      <cell name="de">true</cell>
    </origin>
    <origin name="fr" default="false">
      <cell name="de">false</cell>
      <cell name="fr">true</cell>
    </origin>
    <origin name="uk" default="false">
      <cell name="de">true</cell>
      <cell name="fr">false</cell>
      <cell name="uk">true</cell>
    </origin>
    <origin name="ch" default="false">
      <cell name="ch">true</cell>
    </origin>
    <origin name="sg" default="false">
      <cell name="sg">true</cell>
    </origin>
  </datatype>
  <datatype name="public" default="true">
    <origin name="global" default="true">
      <cell name="uk">false</cell>
    </origin>
    <origin name="eu" default="true">
      <cell name="uk">false</cell>
    </origin>
    <origin name="de" default="true">
      <cell name="uk">false</cell>
    </origin>
    <origin name="fr" default="true">
      <cell name="uk">false</cell>
    </origin>
    <origin name="uk" default="true">
      <cell name="uk">false</cell>
    </origin>
    <origin name="ch" default="true">
      <cell name="uk">false</cell>
    </origin>
    <origin name="sg" default="true">
      <cell name="uk">false</cell>
    </origin>
  </datatype>
</decision>
```

Listing C.4: Policy to allocate backup instances.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<decision xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="celldecisionmatrix.xsd"
          default="true">
  <datatype name="personal" default="false">
    <origin name="global" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="eu" default="false">
      <cell name="de">false</cell>
      <cell name="fr">false</cell>
      <cell name="uk">true</cell>
    </origin>
    <origin name="de" default="false">
      <cell name="de">false</cell>
      <cell name="fr">true</cell>
    </origin>
    <origin name="fr" default="false">
      <cell name="de">true</cell>
      <cell name="fr">false</cell>
      <cell name="uk">false</cell>
    </origin>
    <origin name="uk" default="false">
      <cell name="de">true</cell>
      <cell name="fr">false</cell>
      <cell name="uk">false</cell>
    </origin>
    <origin name="ch" default="false">
      <cell name="ch">true</cell>
    </origin>
    <origin name="sg" default="false">
      <cell name="de">true</cell>
    </origin>
  </datatype>
  <datatype name="financial" default="false">
    <origin name="global" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="eu" default="false">
      <cell name="de">false</cell>
      <cell name="fr">true</cell>
      <cell name="uk">false</cell>
    </origin>
    <origin name="de" default="false">
      <cell name="de">true</cell>
    </origin>
    <origin name="fr" default="false">
      <cell name="fr">true</cell>
    </origin>
    <origin name="uk" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="ch" default="false">
      <cell name="ch">true</cell>
    </origin>
    <origin name="sg" default="false">
      <cell name="sg">true</cell>
    </origin>
  </datatype>
  <datatype name="public" default="false">
    <origin name="global" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="eu" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="de" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="fr" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="uk" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="ch" default="false">
      <cell name="uk">true</cell>
    </origin>
    <origin name="sg" default="false">
      <cell name="de">true</cell>
    </origin>
  </datatype>
</decision>
```

Appendix D

Logs and screenshots of the experiment

In the following, all logged telemetry data and screenshots of the experiment described in Section 6.1.3 are provided.

D.1 Log-files of the nova server

The log-files are created by the NOVA SERVER during the experiment described in Section 6.1.3 and describes the result of the decisions made on basis of the security policies applied (cf. Appendix C.2).

Listing D.1: *Nova Server* log-file of virtual machine no. 1.

```
1  [data]
2  type = "financial"
3  origin = "de"
4  timestamp = "2014-11-27 11:54:36 UTC"
5  [accepted]
6  de = true
7  sg = false
8  fr = false
9  ch = false
10 uk = false
11 api = false
```

ID in OpenStack and file name: 4B59295F-56A9-4126-8C07-4FF0E793F919

Listing D.2: *Nova Server* log-file of back-up instance of virtual machine no. 1.

```
1  [data]
2  type = "financial"
3  origin = "de"
4  timestamp = "2014-11-27 11:54:36 UTC"
5  [accepted]
6  de = true
7  sg = false
8  fr = false
9  ch = false
10 uk = false
11 api = false
```

ID in OpenStack and file name: EA3CC036-87EE-4B70-B5A7-081345A17BDC

Listing D.3: *Nova Server* log-file of virtual machine no. 3.

```
1  [data]
2  type = "financial"
3  origin = "fr"
4  timestamp = "2014-11-27 12:20:14 UTC"
5  [accepted]
6  fr = true
7  sg = false
8  ch = false
9  de = false
10 uk = false
11 api = false
```

ID in OpenStack and file name: A3E635C1-73A3-4263-8CA4-0F6249AA9D34

Listing D.4: *Nova Server* log-file of back-up instance of virtual machine no. 3.

```
1  [data]
2  type = "financial"
3  origin = "fr"
4  timestamp = "2014-11-27 12:20:14 UTC"
5  [accepted]
6  fr = true
7  sg = false
8  ch = false
9  de = false
10 uk = false
11 api = false
```

ID in OpenStack and file name: E827366A-EDB2-4E44-8619-EB8217E5818F

Listing D.5: *Nova Server* log-file of virtual machine no. 4.

```
1  [data]
2  type = "personal"
3  origin = "eu"
4  timestamp = "2014-11-27 12:32:15 UTC"
5  [accepted]
6  fr = true
7  de = true
8  sg = false
9  ch = false
10 uk = false
11 api = false
```

ID in OpenStack and file name: 1ADAFF73-6741-422B-B2F0-7C0E1A2459BD

Listing D.6: *Nova Server* log-file of back-up instance of virtual machine no. 4.

```
1  [data]
2  type = "personal"
3  origin = "eu"
4  timestamp = "2014-11-27 12:32:16 UTC"
5  [accepted]
6  uk = true
7  sg = false
8  fr = false
9  ch = false
10 de = false
11 api = false
```

ID in OpenStack and file name: 6EDED12F-B120-4FC6-8C21-DAA4084630A0

Listing D.7: *Nova Server* log-file of virtual machine no. 2.

```
1  [data]
2  type = "financial"
3  origin = "ch"
4  timestamp = "2014-11-27 12:05:51 UTC"
5  [accepted]
6  ch = true
7  sg = false
8  fr = false
9  de = false
10 uk = false
11 api = false
```

ID in OpenStack and file name: BDB8E85E-0C6D-4AC6-83FB-77BA00AFA5B7

D.2 Logging data of ceilometer

In the following, the telemetry data logged by the CEILOMETER API in OpenStack during the experiment described in Section 6.1.3 are provided. All telemetry data are described by JSON-objects consisting of key-value pairs.

Listing D.8: Ceilometer log for DE COMPUTE 2 before requesting virtual machine no. 1.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d2f79-a15c-8e89-46cc-5084599315e5
  [timestamp] => 2014-11-27T11:47:27
  [recorded_at] => 2014-11-27T11:47:24.987000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => de
      [compute_node] => DeCompute2
      [cpu_features] => hypervisor, osxsave, xsave, ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.amount] => 0
      [memory] => 4048256 KiB
      [node_identifier] => 564d2f79-a15c-8e89-46cc-5084599315e5
      [security_models] => apparmor, dac
      [signature] => <removed signature value>
      [signature_host_identifier] => 564d2f79-a15c-8e89-46cc-5084599315e5
      [signature_pubkey] => <removed public key>
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 2f 79 a1 5c 8e 89-46 cc 50 84 59 93 15 e5
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 291b3bc6-762b-11e4-b512-000c299315e5
  [counter_type] => gauge
)
```

DE COMPUTE 2 is running extended protocol on authentication and integrity. The values of the keys SIGNATURE and SIGNATURE_PUBKEY has been removed due to lack of space.

Listing D.9: Ceilometer log for DE COMPUTE 2 after requesting virtual machine no. 1.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d2f79-a15c-8e89-46cc-5084599315e5
  [timestamp] => 2014-11-27T11:57:28
  [recorded_at] => 2014-11-27T11:57:25.554000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => de
      [compute_node] => DeCompute2
      [cpu_features] => hypervisor, osxsave, xsave, ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.0] => 4b59295f-56a9-4126-8c07-4ff0e793f919
      [instances.amount] => 1
      [memory] => 4048256 KiB
      [node_identifier] => 564d2f79-a15c-8e89-46cc-5084599315e5
      [security_models] => apparmor, dac
      [signature] => <removed signature value>
      [signature_host_identifier] => 564d2f79-a15c-8e89-46cc-5084599315e5
      [signature_pubkey] => <removed public key>
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 2f 79 a1 5c 8e 89-46 cc 50 84 59 93 15 e5
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 8f73f146-762c-11e4-b512-000c299315e5
  [counter_type] => gauge
)
```

DE COMPUTE 2 is running extended protocol on authentication and integrity. The values of the keys SIGNATURE and SIGNATURE_PUBKEY has been removed due to lack of space.

Listing D.10: Ceilometer log for DE COMPUTE 1 before requesting virtual machine no. 1.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d522a-0d36-ac84-dac2-
      e55915e40dd6
  [timestamp] => 2014-11-27T11:47:18
  [recorded_at] => 2014-11-27T11:47:15.484000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => de
      [compute_node] => DeCompute1
      [cpu_features] => hypervisor, osxsave, xsave,
          ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.amount] => 0
      [memory] => 4048352 KiB
      [node_identifier] => 564d522a-0d36-ac84-dac2-
          e55915e40dd6
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 52 2a 0d 36 ac
          84-da c2 e5 59 15 e4 0d d6
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 240ff1ee-762b-11e4-86e9-000
      c29e40dd6
  [counter_type] => gauge
)
```

Listing D.11: Ceilometer log for DE COMPUTE 1 after requesting virtual machine no. 1.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d522a-0d36-ac84-dac2-
      e55915e40dd6
  [timestamp] => 2014-11-27T11:57:19
  [recorded_at] => 2014-11-27T11:57:15.697000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => de
      [compute_node] => DeCompute1
      [cpu_features] => hypervisor, osxsave, xsave,
          ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.0] => ea3cc036-87ee-4b70-b5a7-081345
          a17bdc
      [instances.amount] => 1
      [memory] => 4048352 KiB
      [node_identifier] => 564d522a-0d36-ac84-dac2-
          e55915e40dd6
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 52 2a 0d 36 ac
          84-da c2 e5 59 15 e4 0d d6
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 89dccf32-762c-11e4-86e9-000
      c29e40dd6
  [counter_type] => gauge
)
```

Listing D.12: Ceilometer log for CH COMPUTE 2 before requesting virtual machine no. 2.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564dd34c-03c3-0448-db0c-
      db3fced6ed57
  [timestamp] => 2014-11-27T11:56:26
  [recorded_at] => 2014-11-27T11:56:23.461000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => ch
      [compute_node] => ChCompute2
      [cpu_features] => hypervisor, osxsave, xsave,
          ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.amount] => 0
      [memory] => 4048256 KiB
      [node_identifier] => 564dd34c-03c3-0448-db0c-
          db3fced6ed57
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d d3 4c 03 c3 04
          48-db 0c db 3f ce d6 ed 57
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 6ab7b6a8-762c-11e4-ab7f-000
      c29d6ed57
  [counter_type] => gauge
)
```

Listing D.13: Ceilometer log for CH COMPUTE 2 after requesting virtual machine no. 2.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564dd34c-03c3-0448-db0c-
      db3fced6ed57
  [timestamp] => 2014-11-27T12:06:27
  [recorded_at] => 2014-11-27T12:06:23.921000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => ch
      [compute_node] => ChCompute2
      [cpu_features] => hypervisor, osxsave, xsave,
          ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.0] => bdb8e85e-0c6d-4ac6-83fb-77
          ba00afa5b7
      [instances.amount] => 1
      [memory] => 4048256 KiB
      [node_identifier] => 564dd34c-03c3-0448-db0c-
          db3fced6ed57
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d d3 4c 03 c3 04
          48-db 0c db 3f ce d6 ed 57
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => d0a592ea-762d-11e4-ab7f-000
      c29d6ed57
  [counter_type] => gauge
)
```

Listing D.14: Ceilometer log for FR COMPUTE 2 before requesting virtual machine no. 3.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d7cb4-ac36-0305-dab8-98
      d3d9d32dec
  [timestamp] => 2014-11-27T12:18:30
  [recorded_at] => 2014-11-27T12:18:26.982000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => fr
      [compute_node] => FrCompute2
      [cpu_features] => hypervisor, osxsave, xsave,
          ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.amount] => 0
      [memory] => 4048256 KiB
      [node_identifier] => 564d7cb4-ac36-0305-dab8-98
          d3d9d32dec
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 7c b4 ac 36 03
          05-da b8 98 d3 d9 d3 2d ec
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 7fa33c60-762f-11e4-a713-000
      c29d32dec
  [counter_type] => gauge
)
```

Listing D.15: Ceilometer log for FR COMPUTE 2 after requesting virtual machine no. 3.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d7cb4-ac36-0305-dab8-98
      d3d9d32dec
  [timestamp] => 2014-11-27T12:28:31
  [recorded_at] => 2014-11-27T12:28:27.489000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => fr
      [compute_node] => FrCompute2
      [cpu_features] => hypervisor, osxsave, xsave,
          ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.0] => a3e635c1-73a3-4263-8ca4-0
          f6249aa9d34
      [instances.amount] => 1
      [memory] => 4048256 KiB
      [node_identifier] => 564d7cb4-ac36-0305-dab8-98
          d3d9d32dec
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 7c b4 ac 36 03
          05-da b8 98 d3 d9 d3 2d ec
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => e5974ede-7630-11e4-a713-000
      c29d32dec
  [counter_type] => gauge
)
```

Listing D.16: Ceilometer log for FR COMPUTE 1 before requesting virtual machine no. 3.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d8305-76a9-030a-4f13-a66be29c2ac1
  [timestamp] => 2014-11-27T12:17:53
  [recorded_at] => 2014-11-27T12:17:50.276000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => fr
      [compute_node] => FrCompute1
      [cpu_features] => hypervisor, osxsave, xsave, ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.amount] => 0
      [memory] => 4048256 KiB
      [node_identifier] => 564d8305-76a9-030a-4f13-a66be29c2ac1
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 83 05 76 a9 03 0a-4f 13 a6 6b e2 9c 2a c1
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 69c22fbe-762f-11e4-9096-000c299c2ac1
  [counter_type] => gauge
)
```

Listing D.17: Ceilometer log for FR COMPUTE 1 after requesting virtual machine no. 3.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d8305-76a9-030a-4f13-a66be29c2ac1
  [timestamp] => 2014-11-27T12:27:54
  [recorded_at] => 2014-11-27T12:27:51.375000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => fr
      [compute_node] => FrCompute1
      [cpu_features] => hypervisor, osxsave, xsave, ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.0] => e827366a-edb2-4e44-8619-eb8217e5818f
      [instances.amount] => 1
      [memory] => 4048256 KiB
      [node_identifier] => 564d8305-76a9-030a-4f13-a66be29c2ac1
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 83 05 76 a9 03 0a-4f 13 a6 6b e2 9c 2a c1
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => d01196e6-7630-11e4-9096-000c299c2ac1
  [counter_type] => gauge
)
```

Listing D.18: Ceilometer log for DE COMPUTE 2 before requesting virtual machine no. 4.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d2f79-a15c-8e89-46cc-5084599315e5
  [timestamp] => 2014-11-27T12:27:28
  [recorded_at] => 2014-11-27T12:27:25.140000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => de
      [compute_node] => DeCompute2
      [cpu_features] => hypervisor, osxsave, xsave, ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.0] => 4b59295f-56a9-4126-8c07-4ff0e793f919
      [instances.amount] => 1
      [memory] => 4048256 KiB
      [node_identifier] => 564d2f79-a15c-8e89-46cc-5084599315e5
      [security_models] => apparmor, dac
      [signature] => <removed signature value>
      [signature_host_identifier] => 564d2f79-a15c-8e89-46cc-5084599315e5
      [signature_pubkey] => <removed public key>
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 2f 79 a1 5c 8e 89-46 cc 50 84 59 93 15 e5
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => c00f8186-7630-11e4-b512-000c299315e5
  [counter_type] => gauge
)
```

DE COMPUTE 2 is running extended protocol on authentication and integrity. The values of the keys SIGNATURE and SIGNATURE_PUBKEY has been removed due to lack of space.

Listing D.19: Ceilometer log for DE COMPUTE 2 after requesting virtual machine no. 4.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d2f79-a15c-8e89-46cc-5084599315e5
  [timestamp] => 2014-11-27T12:37:28
  [recorded_at] => 2014-11-27T12:37:25.717000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => de
      [compute_node] => DeCompute2
      [cpu_features] => hypervisor, osxsave, xsave, ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.0] => 4b59295f-56a9-4126-8c07-4ff0e793f919
      [instances.1] => 1adaff73-6741-422b-b2f0-7c0e1a2459bd
      [instances.amount] => 2
      [memory] => 4048256 KiB
      [node_identifier] => 564d2f79-a15c-8e89-46cc-5084599315e5
      [security_models] => apparmor, dac
      [signature] => <removed signature value>
      [signature_host_identifier] => 564d2f79-a15c-8e89-46cc-5084599315e5
      [signature_pubkey] => <removed public key>
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 2f 79 a1 5c 8e 89-46 cc 50 84 59 93 15 e5
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 25f42bd6-7632-11e4-b512-000c299315e5
  [counter_type] => gauge
)
```

DE COMPUTE 2 is running extended protocol on authentication and integrity. The values of the keys SIGNATURE and SIGNATURE_PUBKEY has been removed due to lack of space.

Listing D.20: Ceilometer log for UK COMPUTE 1 before requesting virtual machine no. 4.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d93e9-8a7c-f7af-bcff-814
      f9ff4283b
  [timestamp] => 2014-11-27T12:29:24
  [recorded_at] => 2014-11-27T12:29:21.083000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
    [cell] => uk
      [compute_node] => UkCompute1
      [cpu_features] => hypervisor, osxsave, xsave,
          ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.amount] => 0
      [memory] => 4048256 KiB
      [node_identifier] => 564d93e9-8a7c-f7af-bcff
          -814f9ff4283b
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 93 e9 8a 7c f7
          af-bc ff 81 4f 9f f4 28 3b
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 0586a5d2-7631-11e4-a7dd-000
      c29f4283b
  [counter_type] => gauge
)
```

Listing D.21: Ceilometer log for UK COMPUTE 1 after requesting virtual machine no. 4.

```
(
  [is_authentic] =>
  [counter_name] => host_data
  [user_id] => -
  [resource_id] => 564d93e9-8a7c-f7af-bcff-814
      f9ff4283b
  [timestamp] => 2014-11-27T12:39:25
  [recorded_at] => 2014-11-27T12:39:21.593000
  [resource_metadata] => Array
    (
      [architecture] => x86_64
      [cell] => uk
      [compute_node] => UkCompute1
      [cpu_features] => hypervisor, osxsave, xsave,
          ht, ss, ds, vme
      [cpu_type.cores] => 4
      [cpu_type.model] => Penryn
      [cpu_type.sockets] => 1
      [cpu_type.threads] => 1
      [cpu_type.vendor] => Intel
      [instances.0] => 6eded12f-b120-4fc6-8c21-
          daa4084630a0
      [instances.amount] => 1
      [memory] => 4048256 KiB
      [node_identifier] => 564d93e9-8a7c-f7af-bcff
          -814f9ff4283b
      [security_models] => apparmor, dac
      [sysinfo.manufacturer] => VMware, Inc.
      [sysinfo.product] => VMware Virtual Platform
      [sysinfo.serial] => VMware-56 4d 93 e9 8a 7c f7
          af-bc ff 81 4f 9f f4 28 3b
      [sysinfo.version] => None
    )
  [source] => openstack
  [counter_unit] => host
  [counter_volume] => 1
  [project_id] => -
  [message_id] => 6b78e16a-7632-11e4-a7dd-000
      c29f4283b
  [counter_type] => gauge
)
```

D.3 Screenshots of the dashboard

In the following, the screenshots which were made of the dashboard during the experiment described in Section 6.1.3 are provided.

Figure D.1: Configuration of virtual machine no. 1.

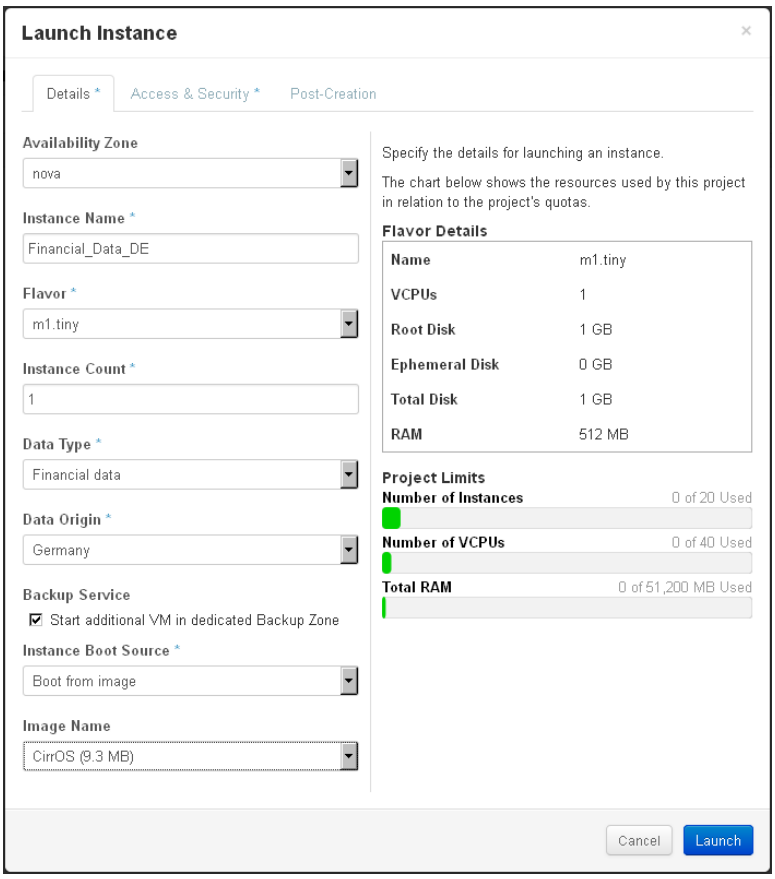


Figure D.2: Running instances after requesting virtual machine no. 1.

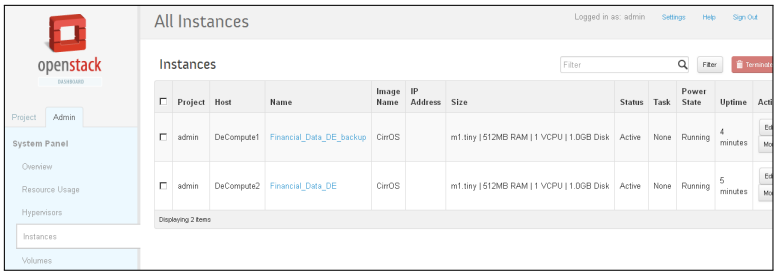


Figure D.3: Running instances after requesting all four virtual machines.

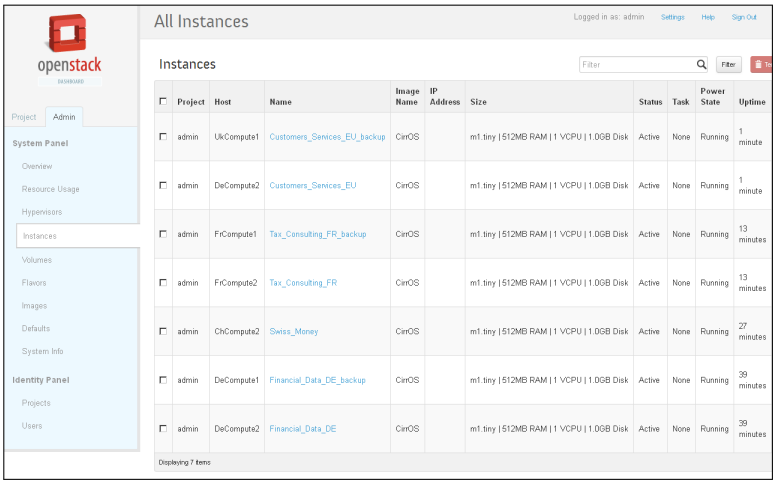
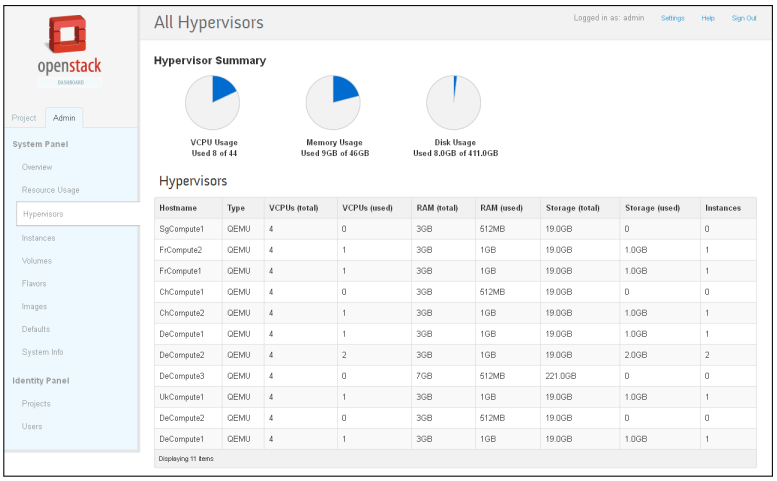


Figure D.4: Hypervisor overview after requesting all four virtual machines.



D.4 Screenshots of the audit board

In the following, the screenshots which were made of the *Analytics Board* during the experiment described in Section 6.1.3 are provided.

Figure D.5: Cloud provider's view (i.e., *Analytics Board*) after requesting all four virtual machines.

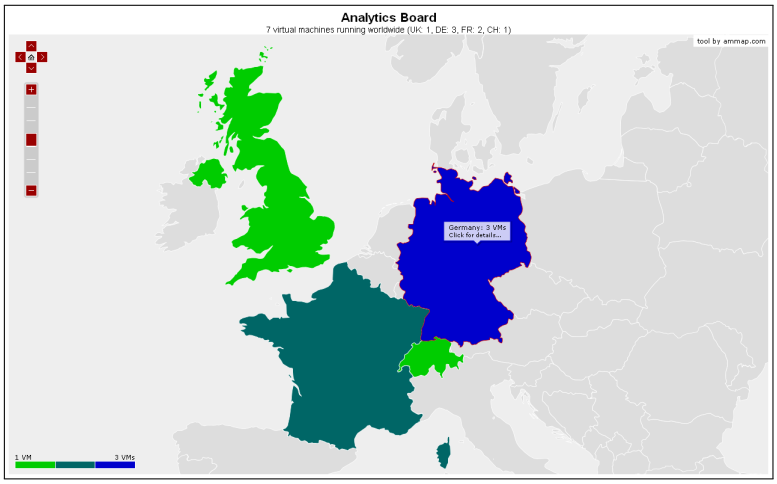


Figure D.6: Details on hosted virtual machine in DE Cell.

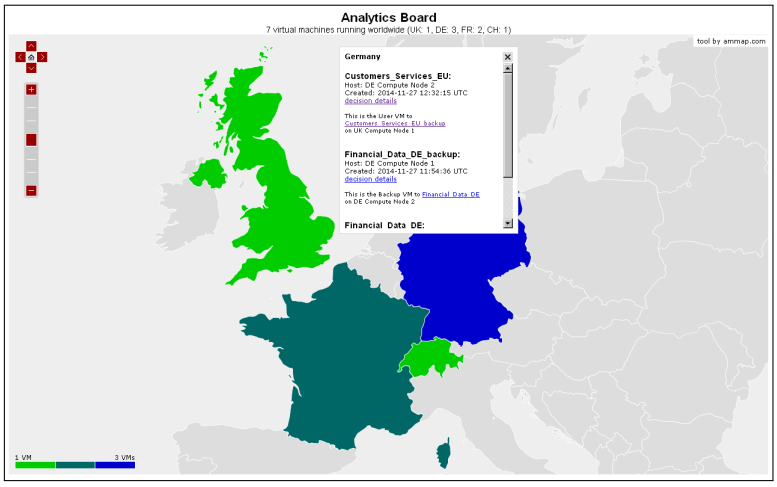


Figure D.7: Visualisation on decision details for virtual machine no. 4.

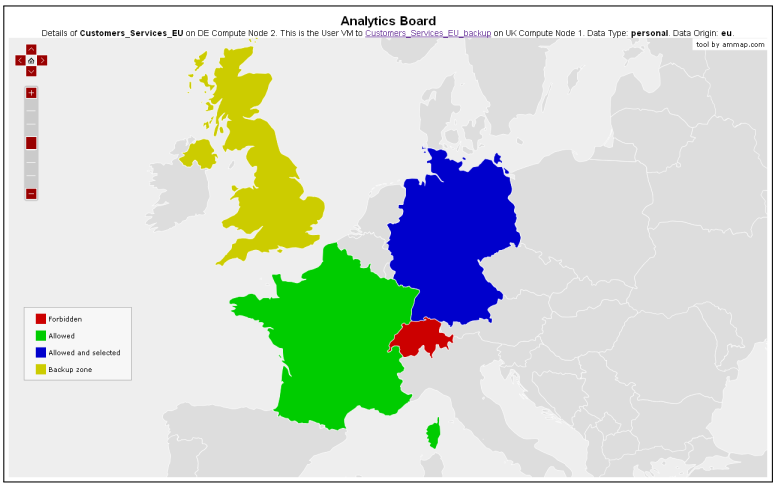
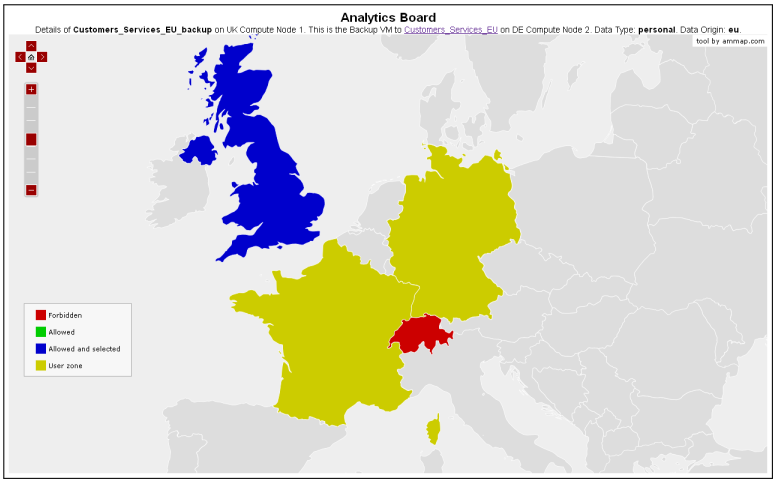


Figure D.8: Visualisation on decision details for back-up instance of virtual machine no. 4.



Glossary

accessing of a virtual resource *[inf.]* a virtual resource is accessed on the virtualisation level (and not only via network connection; cf. Remark 5.1). 117

adequate level of protection *[legal]* ensured level of protection by a country is adequate for admissible data transfer (Art. 25 Data Protection Directive). 2–4, 29, 34, 39–41, 44, 45, 52, 57, 58, 60–63, 65, 95, 180, 186, 194, 196

connecting with a virtual resource *[inf.]* establishing a network connection with a network endpoint of the virtual machine (cf. Remark 5.1). 117

deletion period *[legal]* time period in which stored data has to be deleted. 54, 56

effective legal framework conditions *[legal]* set of conditions of corresponding legal frameworks applying when processing a specific set of data (cf. Def. 3.1). 63

effective level of security *[legal]* set of security measures implemented at a specific location of data processing (cf. Def. 3.2). 63, 89, 91–93, 101, 108, 110, 163, 164, 172, 174, 187, 188, 190, 191

ensured legal framework conditions *[legal]* set of conditions ensured by the legal framework at a specific location of data processing (cf. Def. 3.1). 63, 65, 89

ensured level of security *[legal]* set of security measures obligated by *ensured legal framework conditions* and, additionally, applicable contractual agreements (cf. Def. 3.2). 63, 65, 66, 91, 93, 94, 96, 103, 105, 118, 119, 149–151, 162, 163, 188

information flow of processed data *[inf.]* the access of an actor (cf. Section 2.2.1), virtual resource (cf. Section 4.1.2), or hardware resource (cf. Section 4.1.3) to processed data (cf. Def. 5.1). 117–124, 173, 189, 190, 194, 197, 198

information flow of virtual resources *[inf.]* the access of an actor (cf. Section 2.2.1) or hardware resource (cf. Section 4.1.3) to virtual resources (cf. Def. 5.1). 117–125, 154, 156, 173, 186, 189, 190, 193–199

necessary legal framework conditions *[legal]* set of conditions according to corresponding legal frameworks that has to be ensure by legal frameworks at locations of data processing (cf. Def. 3.1). 63, 191, 194

necessary level of security *[legal]* set of security measures that has to be implemented according to *effective legal framework conditions* and contractual agreements (cf. Def. 3.2). 9, 63–66, 89, 91–96, 101, 103, 105, 111, 112, 118, 121, 123, 124, 149, 151, 162, 163, 174, 185, 186, 191, 192, 194

provision obligation *[legal]* obligation to kept data available and grant access (e.g., for inspections by competent supervisory authorities). 54

purpose limitation *[legal]* collection and processing of personal data are only allowed for ”specified, explicit and legitimate purposes” (Art. 6 para. 1 lit. b Data Protection Directive). 33, 34, 36, 52, 59

retention period *[legal]* time period in which data has to be stored/archived. 54–56, 65

Acronyms

2oo2 2-out-of-2 redundancy. 76, 81

AES Advanced Encryption Standard. 99
AO Abgabenordnung. 3, 22, 32, 34, 42, 43, 48, 49, 54, 55, 60
AWG Außenwirtschaftsgesetz. 50, 51
AWS Amazon Web Service. 2, 3, 70, 71, 75, 76, 83, 201–204
AWV Außenwirtschaftsverordnung. 50

B2B Business-to-Business. 116
BAFA Bundesamt für Wirtschaft und Ausfuhrkontrolle. 51
BDSG Bundesdatenschutzgesetz. 3, 22, 31, 32, 34–38, 40–45, 52, 54, 55, 58
BGBl. Bundesgesetzblatt. 50
BSG Bundessozialgericht. 51

CCITT Comité Consultatif International Téléphonique et Télégraphique. 79
CDMI Cloud Data Management Interface. 98, 101
CIM Common Information Model. 78, 81
CIMI Cloud Infrastructure Management Interface. 83, 96
CMI Cloud Management Interface. 96, 107
CORBA Common Object Request Broker Architecture. 18
CSA Cloud Security Alliance. 11, 64, 66

DAC Discretionary Access Control. 125, 126
DAS Direct Attached Storage. 80
DES Data Encryption Standard. 99
DHCP Dynamic Host Configuration Protocol. 76, 77, 82
DMRZ Deutsches Medizinrechenzentrum. 41
DMTF Distributed Management Task Force. 78
DoS denial of service. 104
DSA Digital Signature Algorithm. 99, 164

ECC error-correcting code. 99, 164
ECtHR European Court of Human Rights. 31
EEA European Economic Area. 31, 47, 59
ENISA European Network and Information Security Agency. 11, 53, 110

EU/EEA European Union/European Economic Area. 3, 4, 6, 21, 22, 29, 39, 46, 50–53, 59, 64, 149–151, 169, 170, 182, 183

FMI Federation Management Interface. 96, 107

GDD Gesellschaft für Datenschutz und Datensicherheit. 37
GDPdU Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen. 48
GDPR General Data Protection Regulation. 31, 36, 37, 45, 59
GG Grundgesetz. 52
GoBD Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff. 10, 42, 43, 48–50, 54, 55
GoBS Grundsätze ordnungsgemäßiger DV-gestützter Buchführungssysteme. 48
GPA Government Procurement Agreement. 52
GPS Global Positioning System. 199
GWB Gesetz gegen Wettbewerbsbeschränkungen. 52

HADDEX Handbuch der deutschen Exportkontrolle. 51
HDD hard disk drive. 80
HDFS Hadoop Distributed File System. 93, 98
HGB Handesgesetzbuch. 3, 4, 48, 55, 60
HTML Hypertext Markup Language. 178
HTTP Hypertext Transfer Protocol. 98, 178
HTTPS Hypertext Transfer Protocol Secure. 98

IaaS Infrastructure-as-a-Service. 2, 4, 8, 14, 15, 17, 18, 27, 33, 35, 47, 48, 67–72, 84, 87, 89, 97, 102, 103, 106, 112, 116, 117, 154, 171, 176, 187, 191, 192, 194, 197, 198, 200, 201, 237
IEEE Institute of Electrical and Electronics Engineers. 98
IP Internet Protocol. 98
IPsec Internet Protocol Security. 97, 98
ISAE International Standard on Assurance Engagements. 21, 111, 172, 190, 191

JSON JavaScript Object Notation. 95, 178, 184, 214

KAGB Kapitalanlagegesetzbuch. 46, 48
KWG Kreditwesengesetz. 3, 34, 46, 47

LAN Local Area Network. 76, 81, 98
LDA Location-Determining resource management and logging Architecture. 10, 175, 176, 178, 179, 182, 184–188, 190, 194

M2M machine-to-machine. 18
MAC Mandatory Access Control. 26, 125, 126
MaRisk Mindestanforderungen an das Risikomanagement. 47

NARA National Archives and Records Administration. 61

NAS network-attached storage. 81, 98
NAT Network Address Translation. 165
NDA non-disclosure agreement. 32, 42
NFS Network File System. 98
NIST National Institute of Standards and Technology. 7, 11, 12, 31, 32, 61, 68, 89, 90, 110, 178, 237

OCCI Open Cloud Computing Interface. 83, 96, 98, 101
OS operating system. 98, 106, 107
OVF Open Virtualization Format. 96

PaaS Platform-as-a-Service. 2, 14, 15, 17, 27, 35, 71, 73, 192, 197, 198, 201
PKI Public Key Infrastructure. 191

QoS Quality of Service. 70, 76, 81, 166

RBAC role-based access control. 101
RPC remote procedure call. 18

SaaS Software-as-a-Service. 2, 14, 15, 17, 35, 192, 197, 198
SAML Security Assertion Markup Language. 100, 101
SAN Storage Area Network. 81, 98
SATA Serial Advanced Technology Attachment. 80
SGB Sozialgesetzbuch. 32, 35, 41–43, 51, 55
SLA service-level agreement. 7, 12, 13, 22, 23, 42, 43, 53, 91, 95, 96, 103, 109–111, 162, 166, 172, 188, 190, 198
SOA service-oriented architecture. 18
SOC Service Organization Controls. 111
SQL Structured Query Language. 178
SSAE Statement on Standards for Attestation Engagements. 21, 111, 190, 191
SSD solid-state drive. 75, 80
SSL Secure Sockets Layer. 98
StGB Strafgesetzbuch. 3, 32, 41, 42, 52
StPo Strafprozeßordnung. 56

TFEU Treaty on the functioning of the European Union. 31
TKG Telekommunikationsgesetz. 31, 32, 42, 43, 54, 55
TLS Transport Layer Security. 98, 101
TMG Telemediengesetz. 31, 55
TPM Trusted Platform Module. 24, 109, 160, 165, 166, 171, 172, 179
TXT Trusted Execution Technology. 179

UPS uninterruptible power supply. 77
USA United States of America. 29, 61, 91, 160, 169, 196
USB Universal Serial Bus. 80
UStG Umsatzsteuergesetz. 49

UWG Gesetz gegen den unlauteren Wettbewerb. 32, 42

VAG Versicherungsaufsichtsgesetz. 46, 47
VLAN Virtual Local Area Network. 81, 82, 98, 181
VMAN Virtualization Management. 96, 98
VMDC Virtualized Multiservice Data Center. 77, 78, 81
VMI Virtualisation Management Interface. 96, 98, 106, 107, 109, 180
VOL/A Vergabe- und Vertragsordnung für Leistungen Teil A. 52
VPLS Virtual Private LAN Service. 98
VPN Virtual Private Network. 76, 97, 98, 106

WpHG Wertpapierhandelsgesetz. 46, 47
WSDL Web Service Description Language. 18
WTO World Trade Organization. 52
WWW World Wide Web. 92, 98

XML Extensible Markup Language. 91, 95, 96, 98, 179, 209
XSD XML Schema Definition. 95, 179, 200, 209

ZAG Zahlungsdiensteaufsichtsgesetz. 46, 47

Symbols

Cloud management process

$\mathbb{BS}$, set of **block storage** instances. 86, 88
$\mathbb{CF}$, set of **compute fabric** instances. 85, 87, 88
$\mathbb{GCM}$, set of **global cluster manager** instances. 84
$\mathbb{GCM}$, set of **local cluster manager** instances. 85
$\overline{cmp} : \mathcal{P}(\mathbb{VR}) \times \mathcal{P}(\mathbb{HW})$, **cloud management process**. 87, 122, 154, 156
$\mathbb{CL}$, set of **cloud infrastructure** instances. 84
$\mathbb{CM}$, set of **compute manager** instances. 79, 84–88
$\mathbb{CN}$, set of **compute node** instances. 86–88
$\mathbb{CSF}$, set of **cloud service fabric** instances. 84, 87, 88
$\mathbb{GC}$, set of **global cluster** instances. 84, 85, 88
$\mathbb{IS}$, set of **image storage** instances. 86
$\mathbb{LC}$, set of **local cluster** instances. 85–88
$\mathbb{NF}$, set of **network fabric** instances. 85, 88
$\mathbb{NM}$, set of **network manager** instances. 85, 86, 88
$\mathbb{NN}$, set of **network node** instances. 86, 88
$\mathbb{OS}$, set of **object storage** instances. 86
$\mathbb{SF}$, set of **storage fabric** instances. 85, 88
$\mathbb{SM}$, set of **storage manager** instances. 85, 86, 88
$\mathbb{SO}$, set of **service orchestrator** instances. 85, 87, 88

Hardware resources

$\mathbb{ADS}$, set of **locally attached data storage** instances. 80
$\mathbb{AMF}$, set of management functions for **accounting management**. 79
$\mathbb{APP}$, set of **applications** hosted by compute server. 80
$\mathbb{CI} := \{\overline{ci}_1, ..., \overline{ci}_n\}$, set of **communication infrastructure** instances $\overline{ci}_i$. 81
$\mathbb{CIS}$, set of **communication infrastructure service** instances. 82, 86–88
$\mathbb{CON}$, set of **end-to-end connection** instances. 82, 86–88
$\mathbb{CON}|\mathbb{P}_{Range}$, set of properties on the **range** of end-to-end connections. 82
$\mathbb{CON}|\mathbb{P}_{Top}$, set of properties on the **communication topology** of end-to-end connections. 82
$\mathbb{CS} := \{\overline{cs}_1, ..., \overline{cs}_n\}$, set of **compute server** instances $\overline{cs}_i$. 79, 86, 88, 156
$\mathbb{DS} := \{\overline{ds}_1, ..., \overline{ds}_n\}$, set of **data storage** instances $\overline{ds}_i$. 80, 81, 86, 88
$\mathbb{FMF}$, set of management functions for **fault management**. 79, 81

List of Figures

List of Tables

List of Listings

List of Definitions

References

[1] Imad M. Abbadi and Anbang Ruan. Towards trustworthy resource scheduling in clouds. *IEEE Transactions on Information Forensics and Security*, 8(6):973–984, Juni 2013. 93

[2] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman. A survey of information-centric networking. *Communications Magazine, IEEE*, 50(7):26–36, 2012. 92

[3] Aiiad Albeshri, Colin Boyd, and Juan Gonzalez Nieto. Geoproof: proofs of geographic location for cloud computing environment. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pages 506–514. IEEE, 2012. 26, 93

[4] Aiiad Albeshri, Colin Boyd, and Juan González Nieto. Enhanced geoproof: improved geographic assurance for data in the cloud. *International Journal of Information Security*, 13:191–198, 2014. 26, 93

[5] *AWS Documentation*. Amazon Web Services, 2014. URL `http://aws.amazon.com/documentation/`. Last visited: 30.06.2015. 2, 71, 99, 101, 201, 202, 203, 204

[6] *Trust Services Principles, Criteria, and Illustrations*. American Institute of Certified Public Accountants (AICPA), 2009. 191

[7] *Statement on Standards for Attestation Engagements (SSAE) No. 16, Reporting on Controls at a Service Organization*. American Institute of Certified Public Accountants (AICPA), 2010. 111, 190, 191

[8] *Reporting on Controls at a Service Organization – Relevant to Security, Availability, Processing, Integrity, Confidentiality, or Privacy (SOC 2)*. American Institute of Certified Public Accountants (AICPA), 2011. 35, 111

[9] Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, and Ming Xu. Web services agreement specification (ws-agreement). In *Open Grid Forum*, volume 128, 2007. 93, 95

[10] *HDFS Users Guide (Version 2.3.0)*. The Apache Software Foundation, 2014. URL `http://hadoop.apache.org/docs/r2.3.0/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html`. Last visited: 30.06.2015. 93, 98

[11] *Opinion 4/2007 on the concept of personal data (01248/07/EN WP 136)*. Articel 29 Data Protection Working Party, 2007. 31

[12] *Opinion 05/2014 on Anonymisation Techniques (0829/14/EN WP216)*. Articel 29 Data Protection Working Party, 2014. 32

[13] Jean Bacon, David Eyers, T Pasquier, Jatinder Singh, Ioannis Papagiannis, and Peter Pietzuch. Information flow control for secure cloud computing. *Network and Service Management, IEEE Transactions on*, PP(99):1–14, 2013. 26, 27, 197

[14] Mirza Basim Baig, Connor Fitzsimons, Suryanarayanan Balasubramanian, Radu Sion, and Donald E Porter. Cloudflow: Cloud-wide policy enforcement using fast vm introspection. In *In Proceedings of the 2nd IEEE International Conference on Cloud Engineering (IC2E 2014)*, 2014. 26, 92

[15] Sundeep Bajikar. *Trusted platform module (tpm) based security on notebook pcs-white paper*. Intel Corporation, 2002. White Paper, Mobile Platforms Group. 165, 172

[16] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007. 91

[17] Erin K. Banks, Michael Bartock, Kevin Fiftal, David Lemon, Karen Scarfone, Uttam Shetty, Murugiah Souppaya, Tarik Williams, and Raghuram Yeluri. *Trusted Geolocation in the Cloud: Proof of Concept Implementation (Draft)*. National Institute of Standards and Technology (NIST), Dezember 2012. NIST Interagency Report 7904 (Draft). 178

[18] Mark Bartel, John Boyer, Barb Fox, Brian LaMacchia, and Ed Simon. Xml-signature syntax and processing. *W3C recommendation*, 12:2002, 2002. 91, 98

[19] Markus Barth. Ortsbestimmtes Cloud-Computing in OpenStack. Master's thesis, University of Passau, 2013. In German. 178, 179, 209

[20] D Elliott Bell and Leonard J La Padula. Secure computer systems: mathematical foundations. Technical Report MTR-2547, Vol. I, The MITRE Corporation, 1973. 9, 27, 124, 126, 127, 128, 134, 135, 136, 138, 140, 144, 145, 155, 156, 174, 199

[21] Stefan Berger, Ramón Cáceres, Dimitrios Pendarakis, Reiner Sailer, Enriquillo Valdez, Ronald Perez, Wayne Schildhauer, and Deepa Srinivasan. Tvdc: managing security in the trusted virtual datacenter. *ACM SIGOPS Operating Systems Review*, 42(1):40–47, 2008. 104

[22] Kenneth J Biba. Integrity considerations for secure computer systems. Technical report, DTIC Document, 1977. 9, 27, 126, 140, 145, 174

[23] Christof Blauberger. Visualisierung von Nachweisdokumentation zur Compliance-Prüfung am Beispiel von IaaS und OpenStack. Bachelor's thesis, University of Passau, 2014. In German. 178, 180

[24] Miguel L Bote-Lorenzo, Yannis A Dimitriadis, and Eduardo Gómez-Sánchez. Grid characteristics and uses: a grid definition. In *Grid Computing*, pages 291–298. Springer, 2004. 17

[25] Ivona Brandic, Vincent C Emeakaroha, Michael Maurer, Schahram Dustdar, Sandor Acs, Attila Kertesz, and Gabor Kecskemeti. Laysi: A layered approach for sla-violation propagation in self-manageable cloud infrastructures. In *Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual*, pages 365–370. IEEE, 2010. 111, 172

[26] Peter Bräutigam, editor. *IT-Outsourcing und Cloud-Computing*. Erich Schmidt Verlag, Berlin, 3. edition, 2013. In German. 34, 35

[27] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. World Wide Web Consortium (W3C), 2008. URL http://www.w3.org/TR/2008/REC-xml-20081126/. Last visited: 30.06.2015. 95

[28] Kirstin Brennscheidt. *Cloud Computing und Datenschutz*. Nomos Verlagsgesellschaft, 2013. In German. 31, 32, 35

[29] Jens Budszus, Oliver Berthold, Alexander Filip, Sven Polenz, Thomas Probst, and Maren Thiermann. *Orientierungshilfe – Cloud Computing der Arbeitskreise Technik und Medien der Konferenz der Datenschutzbeauftragten des Bundes und der Länder sowie der Arbeitsgruppe Internationaler Datenverkehr des Düsseldorfer Kreises*, version 2.0 edition, 2014. in German. 3

[30] *IT-Grundschutz-Standards (BSI-Standard 100-1, BSI-Standard 100-2, BSI-Standard 100-3, BSI-Standard 100-4)*. Bundesamt für Sicherheit in der Informationstechnik BSI, 2008. In German. 41, 47, 123

[31] *IT-Grundschutz-Kataloge – 13. Ergänzungslieferung*. Bundesamt für Sicherheit in der Informationstechnik BSI, 2013. In German. 37, 41, 47, 65, 66, 104, 109, 110

[32] *Rundschreiben 10/2012 (BA) - Mindestanforderungen an das Risikomanagement*. Bundesanstalt für Finanzdienstleistungsaufsicht, 2012. In German. 47

[33] *Rundschreiben 11/2014 (BA) - Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff (GoBD)*. Bundesministerium der Finanzen, 2014. In German. 48

[34] Scott Cantor, Jeff Hodges, John Kemp, Peter Thompson, and Thomas (Ed.) Wason. Liberty id-ff architecture overview. Technical report, Liberty Alliance Project, 2003. 100

[35] Scott Cantor, John Kemp, Rob Philpott, and Eve Maler. *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0*. The Organization for the Advancement of Structured Information Standards (OASIS), 2005. 100

[36] Glenn Carl, George Kesidis, Richard R Brooks, and Suresh Rai. Denial-of-service attack-detection techniques. *Internet Computing, IEEE*, 10(1):82–89, 2006. 99

[37] Steven Carmody, Marlena Erdos, Keith Hazelton, Walter Hoehn, R. L. Morgan, Tom Scavo, David Wasley, and Scott (Ed.) Cantor. Shibboleth architecture - protocols and profiles. Technical report, Internet2, 2005. 100

[38] Antonio Celesti, Francesco Tusa, Massimo Villari, and Antonio Puliafito. How to enhance cloud architectures to enable cross-federation. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 337–345. IEEE, 2010. 96, 98

[39] Chandra Chekuri and Sanjeev Khanna. On multidimensional packing problems. *SIAM journal on computing*, 33(4):837–851, 2004. 189

[40] Ann Chervenak, Vivekenand Vellanki, and Zachary Kurmas. Protecting file systems: A survey of backup techniques. In *Joint NASA and IEEE Mass Storage Conference*, 1998. 99

[41] Ajaegbu Chigozirim et al. Towards building a secure cloud computing environment. *International Journal of Advanced Research in Computer Science*, 3(4):166–171, 2012. 26, 89, 90, 91, 237

[42] Taehwan Choi and Mohamed G Gouda. Http integrity: A lite and secure web against world wide woes. Tech. Rep. TR09-41, Department of Computer Science, The University of Texas at Austin, 2009. 98

[43] Mihai Christodorescu, Reiner Sailer, Douglas Lee Schales, Daniele Sgandurra, and Diego Zamboni. Cloud security is not (just) virtualization security: a short paper. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 97–102. ACM, 2009. 100

[44] *Cisco Virtualized Multi-Tenant Data Center, Version 2.2*. Cisco, 2013. URL `http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/VMDC/2-2/design_guide/vmdcDesign22.pdf`. 77, 78, 81, 98, 99

[45] James Clark. *RELAX NG Specification*. The Organization for the Advancement of Structured Information Standards (OASIS), 2001. URL `http://relaxng.org/spec-20011203.html`. Last visited: 30.06.2015. 95

[46] *Security guidance for critical areas of focus in cloud computing V3.0*. Cloud Security Alliance (CSA), 2011. 11, 64, 66

[47] *Cloud Computing Vulnerability Incidents: A Statistical Overview*. Cloud Security Alliance (CSA), August 2012. Revised: March, 2013. 104

[48] Graham Cormode, Divesh Srivastava, Ting Yu, and Qing Zhang. Anonymizing bipartite graph data using safe groupings. *Proceedings of the VLDB Endowment*, 1(1):833–844, 2008. 99

[49] Sanjaya Dahal. Security architecture for cloud computing platform. Master's thesis, KTH Industrial Engineering and Management, 2012. master's thesis. 101

[50] George I Davida, David L Wells, and John B Kam. A database encryption system with subkeys. *ACM Transactions on Database Systems (TODS)*, 6(2):312–328, 1981. 100

[51] Kelley Dempsey, Nirali Shah Chawla, Arnold Johnson, Ronald Johnston, Alicia Clay Jones, Angela Orebaugh, Matthew Scholl, and Kevin Stine. Information security continuous monitoring (iscm) for federal information systems and organizations. *NIST special publication*, 800:137, 2011. 110

[52] Dorothy E. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19(5):236–243, 1976. 9, 27, 126, 134, 135, 136, 138, 139, 142, 145, 146, 174, 194, 242

[53] Martin Deutschmann, Sebastian Ressi, Sören Bleikertz, Norbert Schirmer, Mihai Bucicoiu, Alysson Bessani, Marcel Santos, Paolo Smiraglia, Roberto Sassu, Johannes Behl, Klaus Stengel, Emanuel Nuno Pereira, and Miguel Areias. *D3.3.4 Final Report on Evaluation Activities*. TClouds Project, 2013. 93, 110

[54] *Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol*. Distributed Management Task Force (DMTF), 2012. 83, 96

[55] *Profile to Enable Automated Deployment of OVF Packages*. Distributed Management Task Force (DMTF), 2013. 96, 98

[56] *Open Virtualization Format Specification*. Distributed Management Task Force (DMTF), 2013. 96

[57] *Common Information Model (CIM) Schema: Version 2.40.0*. Distributed Management Task Force (DMTF), 2014. URL `http://dmtf.org/standards/cim/cim_schema_v2400`. Last visited: 30.06.2015. 78, 81

[58] Bernhard M. Doll. Revisionssichere Dokumentation von Cloud-Ressourcen am Beispiel von OpenStack. Bachelor's thesis, University of Passau, 2014. In German. 178, 180, 190, 191

[59] D. Eastlake and T. Hansen. *US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)*. Internet Engineering Task Force (IETF), 2011. RFC 6234. 99, 164

[60] Donald Eastlake. *XML encryption syntax and processing*. World Wide Web Consortium (W3C), 2003. W3C Recommendation. 91, 98

[61] Tewfiq El Maliki and J-M Seigneur. A survey of user-centric identity management technologies. In *Emerging Security Information, Systems, and Technologies, 2007. SecureWare 2007. The International Conference on*, pages 12–17. IEEE, 2007. 101

[62] Vincent C Emeakaroha, Ivona Brandic, Michael Maurer, and Schahram Dustdar. Low level metrics to high level slas-lom2his framework: Bridging the gap between monitored metrics and sla parameters in cloud environments. In *HPCS*, pages 48–54, 2010. 109

[63] Vincent C Emeakaroha, Marco AS Netto, Rodrigo N Calheiros, Ivona Brandic, Rajkumar Buyya, and César AF De Rose. Towards autonomic detection of sla violations in cloud infrastructures. *Future Generation Computer Systems*, 28(7):1017–1029, 2012. 109, 111, 172

[64] Patrícia Takako Endo, Glauco Estácio Gonçalves, Judith Kelner, and Djamel Sadok. A survey on open-source cloud computing solutions. In *Brazilian Symposium on Computer Networks and Distributed Systems*, 2010. 71

[65] *Commission decisions on the adequacy of the protection of personal data in third countries*. European Commision, 2014. URL `http://ec.europa.eu/justice/data-protection/document/international-transfers/adequacy/index_en.htm`. Last visited: 30.06.2015. 58

[66] *The JSON Data Interchange Format*. European Computer Manufacturers Association (ECMA), 1st edition, 2013. Standard ECMA-404. 95, 178

[67] *Critical Cloud Computing – A CIIP perspective on cloud computing services*. European Network and Information Security Agency ENISA, 2012. 3

[68] *Procure Secure – A guide to monitoring of security service levels in cloud contracts*. European Network and Information Security Agency ENISA, 2012. 110

[69] *Good Practice Guide for securely deploying Governmental Clouds*. European Network and Information Security Agency ENISA, 2013. 53

[70] *Cloud Security Incident Reporting – Framework for reporting about major cloud security incidents*. European Network and Information Security Agency ENISA, 2013. 11, 104

[71] Niroshinie Fernando, Seng W Loke, and Wenny Rahayu. Mobile cloud computing: A survey. *Future Generation Computer Systems*, 29(1):84–106, 2013. 91

[72] A. Fischer, A. Fessi, G. Carle, and H. de Meer. Wide-area virtual machine migration as resilience mechanism. In *Reliable Distributed Systems Workshops (SRDSW), 2011 30th IEEE Symposium on*, pages 72–77, Oct 2011. 152

[73] A. Fischer, J.F. Botero, M. Till Beck, H. de Meer, and X. Hesselbach. Virtual network embedding: A survey. *Communications Surveys Tutorials, IEEE*, 15(4):1888–1906, Fourth 2013. ISSN 1553-877X. 166

[74] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08*, pages 1–10. Ieee, 2008. 17

[75] *Data Sheet Capability Overview: Fujitsu Backup as a Service*. Fujitsu, 2012. 99, 202, 203, 204

[76] Martin Gaedke, Johannes Meinecke, and Martin Nussbaumer. A modeling approach to federated identity and access management. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1156–1157. ACM, 2005. 101

[77] Gregory R Ganger, Bruce L Worthington, Robert Y Hou, and Yale N Patt. Disk arrays: high-performance, high-reliability storage subsystems. *Computer*, 27(3):30–36, 1994. 99

[78] Tal Garfinkel and Mendel Rosenblum. When virtual is harder than real: Security challenges in virtual machine based computing environments. In *HotOS*, 2005. 100

[79] Tal Garfinkel, Ben Pfaff, Jim Chow, Mendel Rosenblum, and Dan Boneh. Terra: A virtual machine-based platform for trusted computing. *SIGOPS Oper. Syst. Rev.*, 37(5): 193–206, 2003. ISSN 0163-5980. 24, 99, 110, 123, 164, 171

[80] Tal Garfinkel, Mendel Rosenblum, et al. A virtual machine introspection based architecture for intrusion detection. In *NDSS*, 2003. 100, 123, 164

[81] Thiago AL Genez, Luiz F Bittencourt, and Edmundo RM Madeira. Workflow scheduling for saas/paas cloud providers considering two sla levels. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pages 906–912. IEEE, 2012. 198

[82] *Template Commission pursuant to Section 11 BDSG*. Gesellschaft für Datenschutz und Datensicherheit (GDD), 2009. URL `https://www.gdd.de/aktuelles/news/englischsprachige-muster-zur-auftragsdatenverarbeitung`. Last visited: 30.06.2015. 37

[83] *Template Inspection of the Commissioned Data*. Gesellschaft für Datenschutz und Datensicherheit (GDD), 2009. URL `https://www.gdd.de/aktuelles/news/englischsprachige-muster-zur-auftragsdatenverarbeitung`. Last visited: 30.06.2015. 37

[84] Ali Ghodsi, Scott Shenker, Teemu Koponen, Ankit Singla, Barath Raghavan, and James Wilcox. Information-centric networking: seeing the forest for the trees. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, page 1. ACM, 2011. 92

[85] Damien Giry. *Cryptographic Key Length Recommendation*. BlueKrypt, 2014. URL `http:\www.keylength.com/en/4/`. Last visited: 30.06.2015. 32

[86] Peter Gola and Rudolf Schomerus. *BDSG Bundesdatenschutzgesetz Kommentar*. Verlag C.H. Beck München, 2007. In German. 40, 52

[87] Daniele Gorla and Rosario Pugliese. Resource access and mobility control with dynamic privileges acquisition. In *Automata, Languages and Programming*, pages 119–132. Springer, 2003. 92

[88] Antonios Gouglidis, Ioannis Mavridis, and Vincent C Hu. Security policy verification for multi-domains in cloud systems. *International Journal of Information Security*, 13 (2):97–111, 2014. 200

[89] Bernd Grobauer, Tobias Walloschek, and Elmar Stocker. Understanding cloud computing vulnerabilities. *Security & privacy, IEEE*, 9(2):50–57, 2011. 104

[90] Nikolay Grozev and Rajkumar Buyya. Inter-cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience*, 44(3):369–390, 2014. 26

[91] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-based geolocation of internet hosts. *Networking, IEEE/ACM Transactions on*, 14(6):1219–1232, Dec 2006. ISSN 1063-6692. 165

[92] Zhenhua Guo, Geoffrey Fox, and Mo Zhou. Investigation of data locality and fairness in mapreduce. In *Proceedings of third international workshop on MapReduce and its Applications Date*, pages 25–32. ACM, 2012. 26

[93] Sebastian Haas, Ralph Herkenhöner, Denis Royer, Ammar Alkassar, Hermann de Meer, and Günter Müller. Supporting semi-automated compliance control by a system design based on the concept of separation of concerns. In *Privacy and Identity Management for Life*, pages 120–129. Springer, 2011. 109

[94] Richard W Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950. 99, 164

[95] D. Hardt. *The OAuth 2.0 Authorization Framework*. Internet Engineering Task Force (IETF), 2012. RFC 6749. 100

[96] Laszlo Hars. Discryption: Internal hard-disk encryption for secure storage. *Computer*, 40(6):103–105, 2007. 99

[97] Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina, and Eduardo B Fernandez. An analysis of security issues for cloud computing. *Journal of Internet Services and Applications*, 4(1):1–13, 2013. 104

[98] Dirk Heckmann. *juris PraxisKommentar Internetrecht*, volume 3. juris Saarbrücken, 2011. In German. 52

[99] Raplh Herkenhöner, Meiko Jensen, Henrich Pöhls, and Hermann De Meer. Towards automated processing of the right of access in inter-organizational web service compositions. In *Proceedings of the IEEE 2010 International Workshop on Web Service and Business Process Security*, 2010. 111

[100] Ralph Herkenhoener, Harald Fischer, and Hermann De Meer. Outsourcing im pflegedienst. *Datenschutz und Datensicherheit (DuD)*, 12:870–874, 2011. In German. 41

[101] Peter Herzog. Osstmm 3: The open source security testing methodology manual – contemporary security testing and analysis. Technical report, Institute for Security and Open Mehtodologies (ISECOM), 2010. 108

[102] Marc Hilbert, editor. *Handbuch Cloud Computing*. Verlag Dr. Otto Schmidt KG, 2014. In German. 33, 34, 35, 41, 44, 47, 48, 49, 50, 51, 52, 53, 56, 57, 58, 60

[103] Mark Hoenike and Lutz Hülsdunk. Outsourcing im versicherungs- und gesundheitswesen ohne einwilligung? *MultiMedia und Recht (MMR)*, pages 788–792, 2004. 41, 52

[104] Gerrit Hornung and Stephan Sädtler. Europas wolken—die auswirkungen des entwurfs für eine datenschutz-grundverordnung auf das cloud computing. *Computer und Recht 2012*, 10/2012:638–645, 2012. In German. 31, 36, 37, 59

[105] Amani S Ibrahim, James Hamlyn Hamlyn-harris, and John Grundy. Emerging security challenges of cloud virtual infrastructure. In *In proceeding of: 2010 Asia Pacific Cloud Workshop*. APSEC, 2010. 104

[106] *802.1Q-2011 - IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks*. Institute of Electrical and Electronics Engineers (IEEE), 2011. 98

[107] *International Standard on Assurance Engagements (ISAE) No. 3402, Assurance Reports on Controls at a Service Organization*. International Auditing and Assurance Standards Board (IAASB), International Federation of Accountants (IFAC), 2009. 111, 172, 190, 191

[108] *IEC 61508:2010 – Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE, or E/E/PES)*. International Electrotechnical Commission, 2010. 153

[109] *Information technology – Security techniques – Information security management – Measurement*. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 2009. ISO/IEC 27004:2009. 37, 47, 104

[110] *Information technology – Security techniques – Information security management system implementation guidance*. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 2010. ISO/IEC 27003:2010. 41, 47

[111] *Information technology – Security techniques – Information security risk management*. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 2011. ISO/IEC 27005:2011. 37, 47, 104

[112] *Information technology – Security techniques – Information security management systems – Requirements*. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 2013. ISO/IEC 27001:2013. 41, 47, 64, 109, 110, 123, 164

[113] *Information technology – Security techniques – Code of practice for information security controls*. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 2013. ISO/IEC 27002:2013. 41, 47, 65

[114] *Transmission Control Protocol*. Internet Engineering Task Force (IETF), 1981. RFC 793. 100

[115] Valérie Issarny, Nikolaos Georgantas, Sara Hachem, Apostolos Zarras, Panos Vassiliadist, Marco Autili, Marco Aurélio Gerosa, and Amira Ben Hamida. Service-oriented middleware for the future internet: state of the art and research directions. *Journal of Internet Services and Applications*, 2(1):23–45, 2011. 199

[116] Pramod Jamkhedkar, Jakub Szefer, Diego Perez-Botero, Tianwei Zhang, Gina Triolo, and Ruby B Lee. A framework for realizing security on demand in cloud computing. In *In Proceedings of 5th IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2013), Bristol, UK*, 2013. 110

[117] Wayne Jansen, Timothy Grance, et al. Guidelines on security and privacy in public cloud computing. *NIST special publication*, 800:144, 2011. 26, 89, 90, 91, 237

[118] Kevin Kelpen. Nachweisdokumentation der Ressourcenzuweisung in der Cloud am Beispiel von OpenStack und IaaS. Bachelor's thesis, University of Passau, 2014. In German. 178, 180, 189, 190

[119] S. Kent and K. Seo. *Security Architecture for the Internet Protocol*. Internet Engineering Task Force (IETF), 2005. RFC 4301. 97, 98

[120] Rasib Hassan Khan, Jukka Ylitalo, and Abu Shohel Ahmed. Openid authentication as a service in openstack. In *Information Assurance and Security (IAS), 2011 7th International Conference on*, pages 372–377. IEEE, 2011. 101

[121] K. Kompella and Y. Rekhter. *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*. Internet Engineering Task Force (IETF), 2007. RFC 4761. 98

[122] Michael A Kozuch, Michael P Ryan, Richard Gass, Steven W Schlosser, David O'Hallaron, James Cipar, Elie Krevat, Julio López, Michael Stroucken, and Gregory R Ganger. Tashi: location-aware cluster management. In *Proceedings of the 1st workshop on Automated control for datacenters and clouds*, pages 43–48. ACM, 2009. 26

[123] Hairong Kuang. *Rack-aware Replica Placement (HADOOP-692)*. The Apache Software Foundation, 2006. URL `https://issues.apache.org/jira/browse/HADOOP-692`. Last visited: 30.60.2015. 93

[124] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishan Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, et al. Oceanstore: An architecture for global-scale persistent storage. *ACM Sigplan Notices*, 35(11):190–201, 2000. 99, 165

[125] Ponnurangam Kumaraguru, L Cranor, Jorge Lobo, and Seraphin Calo. A survey of privacy policy languages. In *Workshop on Usable IT Security Management (USM 07): Proceedings of the 3rd Symposium on Usable Privacy and Security, ACM*. Citeseer, 2007. 95

[126] Jens Lambrecht. *Service Description Fujitsu Cloud IaaS Trusted Public S5*. Fujitsu Technology Solutions, v1.3 edition, June 2013. 202, 203, 204

[127] Len LaPadula, Leonard J LaPadula, and D Elliott Bell. Secure computer systems: A mathematical model. Technical Report MTR-2547, Vol. II, The MITRE Corporation, 1996. 126, 127, 128, 129, 130, 131, 132, 133, 134, 139, 148, 155, 156, 206, 242

[128] Lars Larsson, Daniel Henriksson, and Erik Elmroth. Scheduling and monitoring of internally structured services in cloud federations. In *Computers and Communications (ISCC), 2011 IEEE Symposium on*, pages 173–178. IEEE, 2011. 26

[129] Federico Larumbe and Brunilde Sansò. Optimal location of data centers and software components in cloud computing network design. In *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pages 841–844. IEEE Computer Society, 2012. 26

[130] M. Lasserre and V. Kompella. *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*. Internet Engineering Task Force (IETF), 2007. RFC 4762. 98

[131] Donald C. Latham. *Department of Defense Trusted Computer System Evaluation Criteria*. Department of Defense, 1985. 125

[132] Wubin Li. *Algorithms and Systems for Virtual Machine Scheduling in Cloud Infrastructures*. PhD thesis, Umeå University, Department of Computing Science, 2014. 103, 166

[133] Wubin Li, Johan Tordsson, and Erik Elmroth. Virtual machine placement for predictable and time-constrained peak loads. In Kurt Vanmechelen, JÃ¶rn Altmann, and OmerF. Rana, editors, *Economics of Grids, Clouds, Systems, and Services*, volume 7150 of *Lecture Notes in Computer Science*, pages 120–134. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-28674-2. 166

[134] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, and Dawn Leaf. NIST cloud computing reference architecture. *NIST special publication*, 500:292, 2011. 7, 68, 237

[135] Lawrence Loh and N Venkatraman. *Determinants of information technology outsourcing: a cross-sectional analysis*. International Financial Services Research Center, Sloan School of Management, Massachusetts Institute of Technology, 1992. 17

[136] Meriam Mahjoub, Afef Mdhaffar, Riadh Ben Halima, and Mohamed Jmaiel. A comparative study of the current cloud computing technologies and offers. In *Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on*, pages 131–134. IEEE, 2011. 71

[137] Michael Marc Maish and Alexander Seidl. Rechtliche herausforderungen beim cloud computing in der öffentlichen verwaltung. *Verwaltungsblätter für Baden-Württemberg. Boorberg Verlag.*, 1/2012:7–12, 2012. In German. 52, 53

[138] Nagapramod Mandagere, Pin Zhou, Mark A Smith, and Sandeep Uttamchandani. Demystifying data deduplication. In *Proceedings of the ACM/IFIP/USENIX Middleware'08 Conference Companion*, pages 12–17. ACM, 2008. 99

[139] Philippe Massonet, Syed Naqvi, Christophe Ponsard, Joseph Latanicki, Benny Rochwerger, and Massimo Villari. A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 1510–1517. IEEE, 2011. 103, 111

[140] Erika McCallister, Tim Grance, and Karen Scarfone. Guide to protecting the confidentiality of personally identifiable information (pii). *NIST*, 800:122, 2010. 31, 32

[141] Peter Mell and Timothy Grance. The NIST definition of cloud computing. *NIST special publication*, 800(145):7, 2011. 11, 12, 13, 14, 16, 187

[142] Thijs Metsch, Andy Edmonds, R Nyrén, and A Papaspyrou. *Open cloud computing interface–core*. Open Grid Forum, OCCI-WG, 2010. 83, 96, 101

[143] *Windows Azure Documentation Center*. Microsoft, 2014. URL `https://www.windowsazure.com/en-us/documentation/`. Last visited: 30.06.2015. 2, 71, 99, 101, 201, 202, 203, 204

[144] Mirko Montanari, Jun Ho Huh, Derek Dagit, Rakesh B Bobba, and Roy H Campbell. Evidence of log integrity in policy-based security monitoring. In *Dependable Systems and Networks Workshops (DSN-W), 2012 IEEE/IFIP 42nd International Conference on*, pages 1–6. IEEE, 2012. 109

[145] Luc Moreau, Paul Groth, Simon Miles, Javier Vazquez-Salceda, John Ibbotson, Sheng Jiang, Steve Munroe, Omer Rana, Andreas Schreiber, Victor Tan, et al. The provenance of electronic data. *Communications of the ACM*, 51(4):52–58, 2008. 92

[146] Andrew C Myers and Barbara Liskov. Protecting privacy using the decentralized label model. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 9(4): 410–442, 2000. 126

[147] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11, pages 113–124, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1004-8. 99, 164

[148] Thomas Nägele and Sven Jacobs. Rechtsfragen des cloud computing. *Zeitschrift für Urheber- und Medienrecht (ZUM)*, 4:pp. 281–292, 2010. In German. 35

[149] Kara Nance, Brian Hay, and Matt Bishop. Virtual machine introspection - observation or interference? *IEEE Security & Privacy*, 6(5):32–37, 2008. 98, 99

[150] *Data Encryption Standard (DES)*. National Institute of Standards and Technology (NIST), 1999. FIPS PUB 46-3. 99

[151] *Announcing the Advanced Encryption Standard (AES)*. National Institute of Standards and Technology (NIST), 2001. FIPS PUB 197. 99

[152] *Digital Signature Standard (DSS)*. National Institute of Standards and Technology (NIST), 2009. FIPS PUB 186-3. 99, 164

[153] Qun Ni, Elisa Bertino, Jorge Lobo, Carolyn Brodie, Clare-Marie Karat, John Karat, and Alberto Trombeta. Privacy-aware role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 13(3):24, 2010. 101, 103

[154] *Common Object Request Broker Architecture: Core Specification*. Object Management Group, 2004. 18

[155] *Information Systems Security Assessment Framework (ISSAF)*. Open Information Systems Security Group (OISSG), 2006. Draft 0.2.1. 108

[156] *OpenNebula Documentation (Version 4.4.1)*. OpenNebula Project, February 2014. URL `http://docs.opennebula.org/4.4/index.html`. Last visited: 30.06.2015. 71, 98, 101, 201, 202, 203, 204

[157] *OpenStack Documentation*. OpenStack Foundation, 2014. URL `http://docs.openstack.org/`. Last visited: 30.06.2015. 71, 101, 176, 177, 178, 180, 188, 201, 202, 203, 204

[158] Paul N Otto and Annie I Antón. Addressing legal requirements in requirements engineering. In *RE*, pages 5–14, 2007. 95, 109

[159] Paul N Otto and Annie I Antón. Managing legal texts in requirements engineering. In *Design Requirements Engineering: A Ten-Year Perspective*, pages 374–393. Springer, 2009. 95

[160] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An investigation of geographic mapping techniques for internet hosts. *SIGCOMM Comput. Commun. Rev.*, 31(4):173–185, August 2001. ISSN 0146-4833. 165

[161] Balaji Palanisamy, Aameek Singh, Ling Liu, and Bhushan Jain. Purlieus: locality-aware resource allocation for mapreduce in a cloud. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 58. ACM, 2011. 26

[162] Jongse Park, Daewoo Lee, Bokyeong Kim, Jaehyuk Huh, and Seungryoul Maeng. Locality-aware dynamic vm reconfiguration on mapreduce clouds. In *Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing*, pages 27–36. ACM, 2012. 26

[163] Pankesh Patel, Ajith H Ranabahu, and Amit P Sheth. Service level agreement in cloud computing. Kno.e.sis Publications 78, Wright State University, The Ohio Center of Excellence in Knowledge-Enabled Computing, 2009. 95

[164] Ruxandra Stefania Petre. Data mining in cloud computing. *Database Systems Journal*, 3(3):67–71, 2012. 31

[165] Radu Prodan and Simon Ostermann. A survey and taxonomy of infrastructure as a service and web hosting cloud providers. In *Grid Computing, 2009 10th IEEE/ACM International Conference on*, pages 17–25. IEEE, 2009. 70

[166] Indrakshi Ray, Mahendra Kumar, and Lijun Yu. Lrbac: a location-aware role-based access control model. In *Information Systems Security*, pages 147–161. Springer, 2006. 27, 126, 145

[167] *libvirt: The virtualization API*. Red Hat, Inc., 2014. URL `http://libvirt.org/`. Last visited: 30.06.2015. 180

[168] Holger Reibold. *OpenVAS kompakt*. Bomots-Verlag, 2010. In German. 108

[169] Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, and E. Lear. *Address Allocation for Private Internets*. Internet Engineering Task Force (IETF), 1996. RFC 1918. 165

[170] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*, pages 44–51. Ieee, 2009. 70

[171] Benny Rochwerger, David Breitgand, Eliezer Levy, Alex Galis, Kenneth Nagin, Ignacio Martín Llorente, Rubén Montero, Yaron Wolfsthal, Erik Elmroth, Juan Cáceres, et al. The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4):4–1, 2009. 96, 103

[172] Robert Roßbruch. Die schweigepflicht des pflegepersonals. *PflegeRecht - PfLR*, page 4 seqq. and 34 seqq., 1997. URL `https://www.htwsaar.de/sowi/fakultaet/personen/professoren/prof-dr-robert-rossbruch/veroffentlichungen/schweigepflicht.pdf`. In German. Last visited: 30.06.2015. 41, 52

[173] Alexander Rossnagel. Digital signature regulation and european trends. In G. Müller and K. Rannenberg, editors, *Multilateral Security in Communications, Volume 3: Technology, Infrastructure, and Economy.*, volume 3, pages 235–249. Addison Wesley, 1999. 191

[174] Kai Samelin. Modelling secure cloud-computing. Master's thesis, University of Passau, 2012. 200

[175] Ravi S. Sandhu. Lattice-based access control models. *Computer*, 26(11):9–19, 1993. 27, 140, 149, 194

[176] Ravi S Sandhu and Pierangela Samarati. Access control: principle and practice. *Communications Magazine, IEEE*, 32(9):40–48, 1994. 101

[177] Ravi S Sandhu, Edward J Coynek, Hal L Feinsteink, and Charles E Youmank. Role-based access control models yz. *IEEE computer*, 29(2):38–47, 1996. 101

[178] Fernand Lone Sang, Vincent Nicomette, and Yves Deswarte. I/o attacks in intel pc-based architectures and countermeasures. In *SysSec Workshop (SysSec), 2011 First*, pages 19–26. IEEE, 2011. 98, 99

[179] Nuno Santos, Krishna P Gummadi, and Rodrigo Rodrigues. Towards trusted cloud computing. In *Proceedings of the 2009 conference on Hot topics in cloud computing*, pages 3–3. San Diego, California, 2009. 104

[180] Roberto Sassu, Paolo Smiraglia, Gianluca Ramunno, Alexander Buerger, Norbert Schirmer, Alysson Bessani, Marcel Henrique dos Santos, Sören Bleikertz, Zoltan Nagy, Imad M. Abbadi, Anbang Ruad, Johannes Behl, Klaus Stengel, Sven Bugiel, Hugo Hideler, Stefan Nürnberger, Ninja Marnau, Mina Deng, Zheyi Rong, Miguel Areias, Nuno Emanuel Pereira, and Santos Paulo. *D2.4.2 Initial Component Integration, Final API Specification, and First Reference Platform*. TClouds Project, 2012. 26, 93, 105, 109, 164

[181] Christoph L Schuba, Ivan V Krsul, Markus G Kuhn, Eugene H Spafford, Aurobindo Sundaram, and Diego Zamboni. Analysis of a denial of service attack on tcp. In *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*, pages 208–223. IEEE, 1997. 99

[182] Kent E Seamons, Marianne Winslett, Ting Yu, Bryan Smith, Evan Child, Jared Jacobson, Hyrum Mills, and Lina Yu. Requirements for policy languages for trust negotiation. In *Policies for Distributed Systems and Networks, 2002. Proceedings. Third International Workshop on*, pages 68–79. IEEE, 2002. 95

[183] Ghassan Semaan. Designing networks with the optimal availability. In *Optical Fiber communication/National Fiber Optic Engineers Conference, 2008. OFC/NFOEC 2008. Conference on*, pages 1–6. IEEE, 2008. 99

[184] S. Shepler, M. Eisler, and D. Noveck. *Network File System (NFS) Version 4 Minor Version 1 Protocol*. Internet Engineering Task Force (IETF), 2010. RFC 5661. 98

[185] R. Shirey. *Internet Security Glossary, Version 2*. Internet Engineering Task Force (IETF), 2007. RFC 4949. 125

[186] Sarbjeet Singh and Jagpreet Sidhu. A survey of xml-based security standards for handling security requirements of grid and cloud. *International Journal of Engineering & Technology (0975-4024)*, 5(4):3367–3373, 2013. 95

[187] Seokho Son, Gihun Jung, and Sung Chan Jun. A sla-based cloud computing framework: workload and location aware resource allocation to distributed data centers in a cloud. In *The 2012 international conference on parallel and distributed processing techniques and applications (PDPTA2012), Las Vegas, USA (to be appearing)*, 2012. 26

[188] Axel Spies. Cloud computing: Keine personenbezogenen daten bei verschlüsselung. *MMR-Aktuell*, 3, 2011. In German. 32

[189] Mark Stillwell, David Schanzenbach, Frédéric Vivien, and Henri Casanova. Resource allocation algorithms for virtualized service hosting platforms. *Journal of Parallel and Distributed Computing*, 70(9):962–974, 2010. 189

[190] *Cloud Data Management Interface (CDMI) Version 1.0.2*. Storage Networking Industry Association (SNIA), 2012. 98, 101

[191] S Subashini and V Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11, 2011. 26, 89, 90, 91, 197, 237

[192] S Subramanian, G Nitish Krishna, M Kiran Kumar, P Sreesh, and GR Karpagam. An adaptive algorithm for dynamic priority based virtual machine scheduling in cloud. *International Journal of Computer Science Issues (IJCSI)*, 9(6):397–402, 2012. 95

[193] David Sutherland. A model of information. In *Proc. 9th National Computer Security Conference*, pages 175–183. DTIC Document, 1986. 160, 200

[194] *Symantec Managed PKI Service – Cloud-based service to power strong authentication, encryption, and digital signing applications*. Symantec, 2012. Data Sheet: Authentication. 191

[195] David Tancock, Siani Pearson, and Andrew Charlesworth. A privacy impact assessment tool for cloud computing. In *Privacy and Security for Cloud Computing*, pages 73–123. Springer, 2013. 109

[196] Mingdong Tang, Yechun Jiang, Jianxun Liu, and Xiaoqing Liu. Location-aware collaborative filtering for qos-based service recommendation. In *Web Services (ICWS), 2012 IEEE 19th International Conference on*, pages 202–209. IEEE, 2012. 26

[197] *CCITT Recommendation M.3400 – TMN management functions*. Telecommunication Standardization Sector of ITU (ITU-T), 2000. 79

[198] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. Privacy-preserving anonymization of set-valued data. *Proceedings of the VLDB Endowment*, 1(1):115–125, 2008. 99

[199] Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. *XML Schema Part 1: Structures Second Edition*. World Wide Web Consortium (W3C), 2004. URL http://www.w3.org/TR/xmlschema-1/. Last visited: 30.06.2015. 95

[200] Bill Menezes Tim Zimmerman, Andrew Lerner. *Magic Quadrant for the Wired and Wireless LAN Access Infrastructure (26 June 2014 ID:G00261463)*. Gartner, Inc., 2014. URL http://www.gartner.com/technology/reprints.do?id=1-1WEP20F&ct=140630&st=sb. Last viewed: 30.06.2015. 78

[201] Terkel K Tolstrup, Flemming Nielson, and René Rydhof Hansen. Locality-based security policies. In *Formal Aspects in Security and Trust*, pages 185–201. Springer, 2007. 27, 92, 126, 145

[202] Johan Tordsson, Erik Elmroth, Daniel Henriksson, Petter Svärd, Wubin Li, Mina Sedhagat, Ahmed Aley-Eldin, Juan Luis Prieto, Francisco Javier Nieto, Ana Juan Ferrer, J. Oriol Fitó, Mario Macias, Raül Sivent, Jorge Ejarque, Enric Tejedor, Jordi Guitart, Afnan Ullah Khan, Porwal. Sakshi, Srijith Nair, Pramod Pawar, Ali Sajjad, Kleopatra Konstanteli, George Kousiouris, George Vafiadis, Karl Catewicz, Hassan Rasheed, Angela Rumpl, Oliver Wäldrich, Thomas Weuffel, Wolfgang Ziegler, Karim Djemame, Django Armstrong, Ming Jiang, Mariam Kiran, Roland Kübert, Gregory Katsaros, Tinghe Wang, Craig Sheridan, and Tabassum Sharif. D1.2.2.2 — optimis detailed design. Technical report, Optimised Infrastructure Services (OPTIMIS), FP7 Integrated Project, 2011. 26, 92, 105, 110

[203] *TPM Main - Part 1 Design Principles*. Trusted Computing Group (TCG), 2011. Specification Version 1.2, Revision 116. 109

[204] Wei-Tek Tsai, Xin Sun, and Janaka Balasooriya. Service-oriented cloud computing architecture. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, pages 684–689. IEEE, 2010. 83

[205] S. Turner and T. Polk. *Prohibiting Secure Sockets Layer (SSL) Version 2.0*. Internet Engineering Task Force (IETF), 2011. RFC 6176. 98

[206] *Trusted Site Infrastructure*. TÜV Informationstechnik GmbH, 2014. In German. 123, 164

[207] Kazi Wali Ullah, Abu Shohel Ahmed, and Jukka Ylitalo. Towards building an automated security compliance tool for the cloud. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 1587–1593. IEEE, 2013. 108

[208] *EuroPriSe Criteria*. Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein (ULD), November 2011. 37

[209] *GAO Report to Congressional Requesters – Privacy – Alternatives Exist for Enhancing Protection of Personally Identifiable Information (GAO-08-536)*. United States Government Accountability Office, 2008. 31

[210] Hien Nguyen Van, Frederic Dang Tran, and J-M Menaud. Sla-aware virtual resource management for cloud infrastructures. In *Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on*, volume 1, pages 357–362. IEEE, 2009. 166

[211] Luis M Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1):50–55, 2008. 11, 12, 13, 187

[212] Luis M Vaquero, Luis Rodero-Merino, and Daniel Morán. Locking the sky: a survey on iaas cloud security. *Computing*, 91(1):93–118, 2011. 104

[213] Sofie Verbrugge, Didier Colle, Piet Demeester, Ralf Huelsermann, and Monika Jaeger. General availability model for multilayer transport networks. In *Design of Reliable Communication Networks, 2005.(DRCN 2005). Proceedings. 5th International Workshop on*, pages 8–pp. IEEE, 2005. 98

[214] Akshat Verma, Puneet Ahuja, and Anindya Neogi. pmapper: Power and migration cost aware application placement in virtualized systems. In ValÃ©rie Issarny and Richard Schantz, editors, *Middleware 2008*, volume 5346 of *Lecture Notes in Computer Science*, pages 243–264. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-89855-9. 166

[215] Ivan Voras, B Mihaljevic, and M Orlic. Criteria for evaluation of open source cloud computing solutions. In *Information Technology Interfaces (ITI), Proceedings of the ITI 2011 33rd International Conference on*, pages 137–142. IEEE, 2011. 71, 201

[216] John Wack, Miles Tracy, and Murugiah Souppaya. Guideline on network security testing. *Nist special publication*, 800:42, 2003. 108

[217] Karen Waltermire and Mike Bartock. *Trusted Geolocation in The Cloud Technical Demonstration*. National Institute of Standards and Technology (NIST), 2013. Presentation on April 11th, 2014 at the Forum of the National Cybersecurity Center of Excellence (NCCoE). 26, 94, 178, 179

[218] Yong Wang, Daniel Burgener, Marcel Flores, Aleksandar Kuzmanovic, and Cheng Huang. Towards street-level client-independent ip geolocation. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 365–379, Berkeley, CA, USA, 2011. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1972457.1972494. 165

[219] Thilo Weichert. *Inanspruchnahme des Patriot Acts und anderer US-rechtlicher Regelungen zur Beschaffung von personenbezogenen Daten aus dem Raum der Europäischen Union durch US-Behörden*. Unabhängigen Landeszentrums für Datenschutz Schleswig-Holstein (ULD), 2011. URL https://www.datenschutzzentrum.de/internationales/20111115-patriot-act.html. In German. Last visited: 30.06.2015. 29

[220] *Web Services Description Language (WSDL) 1.1*. World Wide Web Consortium (W3C), 2001. 18

[221] Min Xie, Yuan-Shun Dai, and Kim-Leng Poh. *Computing System Reliability*. Springer, 2004. 152, 243

[222] Mingqiang Xue, Panos Kalnis, and Hung Keng Pung. Location diversity: Enhanced privacy protection in location based services. In *Location and Context Awareness*, pages 70–87. Springer, 2009. 25

[223] Mariemma I Yagüe. Survey on xml-based policy languages for open environments. *Journal of Information Assurance and Security*, 1(1):11–20, 2006. 95

[224] Chenyu Yan, Daniel Englender, Milos Prvulovic, Brian Rogers, and Yan Solihin. Improving cost, performance, and security of memory encryption and authentication. *SIGARCH Comput. Archit. News*, 34(2):179–190, 2006. ISSN 0163-5964. 99

[225] George Yee and Larry Korba. Privacy policy compliance for web services. In *Web Services, 2004. Proceedings. IEEE International Conference on*, pages 158–165. IEEE, 2004. 109

[226] I. Youn, B.L. Mark, and D. Richards. Statistical geolocation of internet hosts. In *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, pages 1–6, Aug 2009. 165

[227] Lamia Youseff, Maria Butrico, and Dilma Da Silva. Toward a unified ontology of cloud computing. In *Grid Computing Environments Workshop, 2008. GCE'08*, pages 1–10. IEEE, 2008. 70

[228] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1):7–18, 2010. 70

[229] Jun Zhao and Olaf Hartig. Towards interoperable provenance publication on the linked data web. In *LDOW*, 2012. 92

[230] Minqi Zhou, Rong Zhang, Wei Xie, Weining Qian, and Aoying Zhou. Security and privacy in cloud computing: A survey. In *Semantics Knowledge and Grid (SKG), 2010 Sixth International Conference on*, pages 105–112. IEEE, 2010. 26, 89, 90, 91, 237

[231] Wolfgang Ziegler and Ming Jiang. Optimis sla framework and term languages for slas in cloud environment. Technical report, Optimised Infrastructure Services (OPTIMIS), FP7 Integrated Project, 2011. 93, 95