

# **Pre-Ingest-Werkzeug für Digitalisate aus Kulturerbe- Einrichtungen des Landes Brandenburg**

---

**Konzeption und programmtechnische Umsetzung einer prototypischen  
webbasierten Portallösung**

## **BACHELORARBEIT**

zur Erlangung des akademischen Grades Bachelor of Arts  
im Studiengang Information und Dokumentation  
an der Fachhochschule Potsdam

Fachbereich: Informationswissenschaften

Vorgelegt von

**Eric Neudhöfer**

Matrikelnummer: 12921

### **Gutachter:**

Erstgutachter: Prof. Dr. Günther Neher  
Zweitgutachter: M.A. Ulf Preuß

*Eingereicht am 27.02.2017*

# Inhaltsverzeichnis

Abkürzungsverzeichnis .....	3
<b>1 Einleitung.....</b>	<b>4</b>
<b>2 Herausforderung und Lösungsansatz .....</b>	<b>6</b>
2.1 Herausforderungen in der digitalen Langzeitarchivierung .....	6
2.2 Lösungsansatz in Berlin-Brandenburg .....	8
2.3 Das OAIS-Funktionsmodell .....	9
2.4 INGEST und PRE-INGEST .....	10
2.5 Der Identifikator ISIL.....	11
2.6 Die Systemarchitektur des Demonstrators.....	13
<b>3 Datenbankmodellierung .....</b>	<b>16</b>
3.1 Anforderungen .....	16
3.2 Identifizierung der Datenbasis .....	16
3.3 Untersuchung der Objektbeziehungen .....	19
3.4 Das Entity-Relationship-Modell (ERM) .....	20
3.5 Überprüfung des ER-Modells .....	21
3.6 Transformation von ERM in RDM.....	23
<b>4 Entwicklung der prototypischen Webanwendung .....</b>	<b>25</b>
4.1 Allgemeines und Methodik .....	25
4.2 Verzeichnisstruktur .....	26
4.3 Grundstruktur der Webseiten .....	28
4.4 Programmierung der Webanwendung .....	31
4.5 Simulation des Archivierungsprozesses .....	37
<b>5 Schlussfolgerung und Ausblick .....</b>	<b>39</b>
5.1 Zielsetzung und Lösung .....	39
5.2 Interaktion und Fehlerbehandlung .....	40
5.3 Hohe Datenmengen und Upload .....	40
5.4 Metadaten und modulare Erweiterungen .....	42
<b>Fazit.....</b>	<b>44</b>
Abbildungsverzeichnis .....	45
Tabellenverzeichnis .....	45
Literaturverzeichnis .....	46
Anhang .....	48
Eidesstattliche Erklärung .....	65

## Abkürzungsverzeichnis

Kürzel	Beschreibung
AIP	Archival Information Package
AMD	Administrative Metadaten
CSS	Cascading Style Sheets
CSV	Comma-separated value
digiS	Servicestelle Digitalisierung am Konrad-Zuse-Institut Berlin
DIN	Deutsches Institut für Normung e. V.
DIP	Dissemination Information Package
DMD	Deskriptive Metadaten
ERM	Entity-Relationship-Modell
FTP	File Transfer Protocol
HTML	Hypertext Markup Language
ISIL	International Standard Identifier for Libraries and Related Organizations
ISO	International Organization for Standardization
KBD	Koordinierungsstelle Brandenburg.digital
METS	Metadata Encoding and Transmission Standard
OAIS	Open Archival Information System
PHP	PHP: Hypertext Preprocessor (urspr. Personal Home Page Tools)
RDM	Relationales Datenbankmodell
SIP	Submission Information Package
SQL	Structured Query Language
TIP	Transfer Information Package
TMD	Technische Metadaten
XML	Extensible Markup Language

## 1 Einleitung

Zur langfristigen Erhaltung digitaler Informationen, stehen Einrichtungen, welche sich fortwährend mit dem Thema digitale Langzeitarchivierung auseinandersetzen müssen, vor immer neuen Herausforderungen. Das rasante Voranschreiten von Technologien unterstützt und erschwert es zugleich. Neben der Erfassung und Aufbewahrung von Informationen, stellt vor allem die dauerhafte Verfügbarkeit das größte Problem dar. Stetige Weiterentwicklungen von Software und Hardware gefährden weitgehend Kompatibilitäten der Datenstrukturen und damit auch die Interpretationsmöglichkeiten bereits archivierter Digitalisate. Insbesondere kommunale Einrichtungen handeln diesbezüglich oftmals eigenverantwortlich. Dies erfordert gewisse Kompetenzen der Mitarbeiter sowie technische und finanzielle Ressourcen – Kriterien, welche häufig unzureichend erfüllt sind. Aus Ermangelung einer fachgemäßen digitalen Langzeitarchivierung folgt eine Bedrohung des Verlustes digitaler Informationen. Werden keine Erhaltungsmaßnahmen ergriffen, gehen die Kulturgüter der Einrichtungen unwiderruflich verloren. In Einrichtungen des kulturellen Erbes der Länder Brandenburg und Berlin gibt es zu diesem Thema bereits Planungen einer Verbundlösung. Die Koordinierungsstelle Brandenburg-digital (KBD), welche derartige Organisationen initiierte, möchte in Kooperation mit der Service-stelle Digitalisierung (digiS) spartenübergreifend die digitalen Kulturgüter aus den Ebenen der Kommunen und des Landes archivieren und landes-übergreifende Kooperationen schaffen. Ein Archivsystem für dieses Unternehmen wird unter der Beteiligung der Fachhochschule Potsdam mittels der Software Archivemata prototypisch entwickelt. In der Regel beginnen die Prozesse eines Archivsystems mit dem Ingest, einer funktionellen Transfer-einheit. Entworfenen Modelle erläutern ausführlich den Ingest-Bereich. Die Praxis zeigt hingegen, dass weitere Prozesse vor dieser Funktionseinheit notwendig sind.

Die vorliegende Bachelorarbeit untersucht die Aggregation und die Vor-verarbeitungsprozesse von Digitalisaten, welche noch vor dem Ingest eines Archivsystems stattfinden – dem Pre-Ingest. Dazu wird eine Webanwendung

konzeptioniert und demonstrativ umgesetzt. Es soll analysiert werden, welche Anforderungen an einer solchen Lösung gebunden sind und mit welchen Methoden sie umgesetzt werden kann. Diese Aspekte reichen von der Planung des Designs und Aufbaus eines Demonstrators bis hin zum Datenverkehr zwischen dem Portal und dem Archivsystem.

## 2 Herausforderung und Lösungsansatz

### 2.1 Herausforderungen in der digitalen Langzeitarchivierung

In erster Linie beinhaltet der Begriff der Langzeitarchivierung nach Dobratz und Tappenbeck (2002) nicht die intakte Aufbewahrung digitaler Ressourcen in ihrem ursprünglichen Format, sondern primär die inhaltliche Nutzbarkeit und somit die Langzeitverfügbarkeit der digitalen Dokumente.

Dabei sind die Begriffe „Archivierung“ und „Langzeitarchivierung“ inhaltlich zu unterscheiden. Der Archivierungsbegriff wird in der Informationstechnologie dafür verwendet, um die Aspekte der Speichermechanismen, der Datenformatierung und der Datenerhaltung zu beschreiben. Er nimmt Bezug auf die physikalische Speicherung von Daten, die ebenso für die Langzeitarchivierung erforderlich sind. Demgegenüber bezieht sich der Hauptaspekt der Langzeitarchivierung weniger auf diese Daten, sondern auf die Erhaltung abgebildeter Informationen (Keitel und Schoger, 2013, S. 5-6). Im Allgemeinen birgt die digitale Langzeitarchivierung viele Herausforderungen. Einem einfachen Lösungsansatz stehen dieser Thematik vielfältige Faktoren entgegen. Fragestellungen zu den Möglichkeiten und Bedingungen vertrauenswürdiger Archivierung sind gegenwärtig unzureichend beantwortet. „Dies gilt sowohl für die Sicherung der Datenspeicherung (Trägermedium) als auch den zukünftigen Zugriff auf die in ihnen enthaltenen Informationen (Datenformate) und deren dauerhafte Nutzbarkeit (Erschließung)“ (Dobratz und Tappenbeck, 2002, S. 257). Für technische und organisatorische Lösungsansätze sind insbesondere fehlende oder unzureichend beschriebene Standards hindernd. Umfassende und verbindliche Richtlinien werden für verschiedene Aspekte der digitalen Langzeitarchivierung benötigt, dazu zählen: eine Organisationsstruktur von Archivierungstechniken, eine inhaltliche Dokumentenstrukturierung in Hinblick auf Emulation und Migration und eine strukturierte Verwendung der Metadaten für ein einheitliches Retrieval (Dobratz und Tappenbeck, 2002, S. 257-259).

Der Prozess der Langzeitarchivierung findet in einem digitalen Langzeitarchiv statt. Dabei handelt es sich um eine „Organisation (bestehend aus Personen

und technischen Systemen), die die Verantwortung für den Langzeiterhalt und die Langzeitverfügbarkeit von Informationen in digitaler Form sowie die Bereitstellung für eine bestimmte Zielgruppe übernommen hat“ (DIN 31644, 2012). Innerhalb eines digitalen Langzeitarchivs als technisches System sind Überlegungen und Handlungen bezüglich diverser Problemstellungen erforderlich. Um die Informationen und die Authentizität von Digitalisaten nicht zu verfälschen, sollte auf eine möglichst minimale Konvertierung und auf die Art der Kodierung geachtet werden. Entwicklungen einer sicheren Struktur im technischen und organisatorischen Kontext sind erforderlich, um die Authentizität und Integrität der Daten zu gewährleisten, welche vor allem durch Migrationsprozesse gefährdet sind. Beim Überführungsprozess in ein technisches System sind Standardisierungen und Anforderungen an das Transferformat hilfreich. Festgesetzte Workflows bei Kooperationen verschiedener Akteure sind gleichermaßen sinnvoll. Unter anderem unterliegen die Herausforderungen den im Folgenden erläuterten Aspekten. Die Entwicklung und Konzeption der technischen Archivierung, etwa in Hinblick auf Datenträger und Archivsysteme, ist eine andauernde Aufgabe und Bedarf einer ständigen Marktbeobachtung. Die Daten und die in einem Prozess generierten relevanten Metadaten bilden ein Transfercontainer oder Transferpaket. Dabei benötigt die Verwaltung der Metadaten für eine hohe Kompatibilität ein Strukturmodell. Zudem erhalten rechtliche Fragen in Bezug auf Transaktionen und Modifikationen von Digitalisaten fortdauernd mehr Gewichtung, wodurch das Digital Rights Management in die Entwicklung mit eingebunden werden sollte. Eine weitere Herausforderung stellt die Koordinierung von Initiativen im Bereich der Langzeitarchivierung dar, um Kooperationen zu bilden. Doppelte Entwicklungsprozesse könnten und sollten vermieden werden, um eine hohe Effektivität und Effizienz in der Konzeption und Entwicklung durch Synergie zu schaffen. Eine stetige Beobachtung der Tendenzen in der Standardisierung sollte auf nationaler, aber auch auf internationaler Ebene stattfinden (Dobratz und Tappenbeck, 2002, S. 259-261). Die Herausforderungen einer nachhaltigen Sicherung des kulturellen Erbes sind umfassend: von der Digitalisierung über Medienwechsel bis zur Verantwortung und Finanzierung digitaler Langzeitarchivierung. In Bezug auf die Nachhaltigkeit sollte die Sichtweise der

gesamten Problematik auf verschiedene Aspekte hin erläutert werden. Hierzu zählt nicht nur die technische und archivarische Sicht, sondern auch eine kulturpolitische, sozial-ökologische, rechtliche und finanzielle Perspektive (Klimpel, 2013, S. 11-13). Die digitale Langzeitarchivierung an sich tangiert vielzählige Themenbereiche. Neben der Informations- und Archivwissenschaft, deren Perspektive in dieser Arbeit fokussiert wird, sind unter anderem die Bereiche Bibliothekswissenschaft (bspw. Katalogisierung), Rechtswissenschaft (Urheberrecht beim Kopieren und Archivieren im Internet), Kulturwissenschaft (die Differenzierung zu archivierender oder nicht-archivierender Fragen), Informatik und Informationstechnologie (Massenspeicher und Zugriffssysteme) betroffen. Durch die Komplexität des Themas digitaler Langzeitarchivierung werden Entwicklungen von nationalen Strategien massiv erschwert. Der Verlust digitaler Kulturgüter ist im hohen Maße bedroht – national sowie international (Rehm, 2013, S. 199-201). Trotz der internationalen Herausforderung werden zum Beispiel in Deutschland die Lösungen auf Landesebene entwickelt. In den Bundesländern Brandenburg und Berlin finden derzeit Planungen zur Archivierung kultureller Informationen in einer Verbundlösung statt.

## **2.2 Lösungsansatz in Berlin-Brandenburg**

Um den Verlusten kulturellen Erbes entgegenzuwirken, wird für die Region Berlin-Brandenburg an einer allgemeinen Lösung gearbeitet. Vor allem durch die Eigenverantwortlichkeiten in der Archivierung kommunaler Einrichtungen können Risiken, durch positive Synergieeffekte reduziert werden. Durch eine Verbundlösung soll eine nachhaltige Langzeitarchivierung und Langzeitverfügbarkeit von digitalen Objekten, gegebenenfalls aus Digitalisierungsprojekten, entstehen. Die Initiative dieses Vorhabens geht auf die Koordinierungsstelle Brandenburg-digital (KBD) zurück. Die KBD begleitet Digitalisierungsvorhaben spartenübergreifender Projekte. Des Weiteren gehören Beratung und Aufbau kooperativer Partnerschaften zu den Aufgaben der KBD. In Zusammenarbeit mit der Servicestelle Digitalisierung am Konrad-Zuse-Institut Berlin (digiS) entsteht eine landesübergreifende Verbindung. Technische Lösungsansätze sollen entwickelt werden.



Diesbezüglich wird grundsätzlich auf ein Archivsystem zurückgegriffen. Unter der Beteiligung der Fachhochschule Potsdam wird zur prototypischen Entwicklung einer Archivlösung Archivemata, eine webbasierte Open-Source-Software, angewendet. Mittels dieser Software werden Übernahmepakete und Archivinformationspakete generiert. Hierzu liefert eine Einrichtung als Produzent Digitalisate und Metadaten. Ein automatisierter Archivierungsvorgang findet durch Archivemata statt. Die Funktionsweise der Software richtet sich nach dem OAIS-Referenzmodell (Open Archival Information System). In diesem Modell werden vorverarbeitende Prozesse zum Übernahmepaket unzureichend beschrieben. Die Verbundlösung Berlin-Brandenburg sieht daher eine vorverarbeitende Instanz als Sammelstelle vor. Diese dient als webbasiertes Portal für das Archivsystem als sogenannter Deposit oder auch Pre-Ingest. Das Webportal soll es ermöglichen Datenlieferungen entgegenzunehmen, diese zu überprüfen und wieder auszuliefern. Des Weiteren soll es Identifier generieren, um Datenlieferungen zu identifizieren, Metadaten erfassen und Metadatendateien bereitstellen. Die dadurch entstandenen Daten werden an die erste Funktionseinheit „Ingest“ des OAIS-Modells geliefert beziehungsweise in das Archivsystem überführt.

## 2.3 Das OAIS-Funktionsmodell

Das OAIS-Funktionsmodell wird in der Norm ISO 14721:2012<sup>1</sup> beschrieben. Das Modell ist in sechs Funktionseinheiten gegliedert: Übernahme (Ingest), Datenverwaltung, Archivspeicher, Zugriff, Administration und Erhaltungsplanung. Die Pfade der Kommunikation zwischen den Einheiten werden im Modell durch Linien grafisch dargestellt (siehe Abbildung 1). Das Funktionsmodell beschreibt die Übernahme des Submission Information Package (SIP) vom Produzenten in das Archivsystem. Damit ist das SIP das Übernahmeformationspaket. Innerhalb des Archivsystems wird aus dem SIP ein Archival Information Package (AIP) generiert. Dieses Archivinformationspaket kann nachkommend als Dissemination Information

---

<sup>1</sup> ISO 14721:2012 – Space data and information transfer systems – Open archival information system (OAIS) – Reference model

Package (DIP) vom Endnutzer bestellt und angefragt werden. Das DIP ist infolgedessen das Auslieferungsinformationpaket (nestor-Arbeitsgruppe OAI-Übersetzung/Terminologie, 2013, S. 33-35).

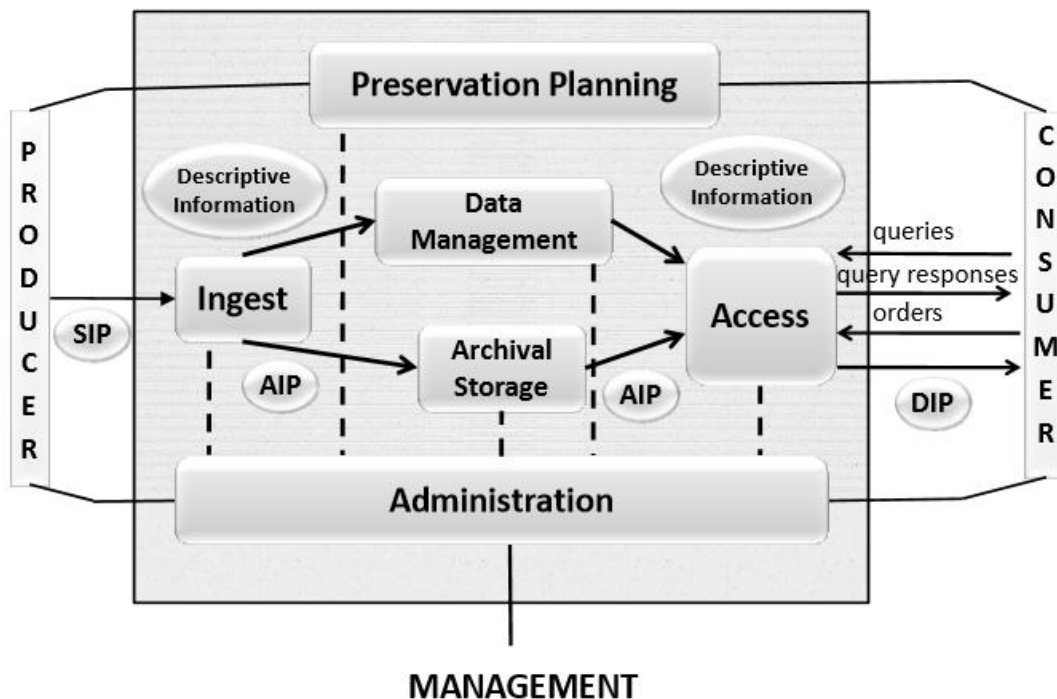


Abbildung 1: OAI-Funktionsmodell nach ISO 14721:2012

Für eine prototypische Webanwendung als Portallösung eines Archivsystems ist zunächst nur die Funktionseinheit Ingest relevant.

## 2.4 INGEST und PRE-INGEST

Der Ingest-Bereich als Funktionseinheit dient zur Übernahme, der – von den Produzenten stammenden – SIPs in das digitale Archiv. Zugleich ist es zuständig für diverse Überprüfungen, um sicherzustellen, dass letztendlich nachhaltige Informationspakete archiviert werden können (Brübach, 2010). Die Gestaltung des SIPs (Datenstruktur, Format, etc.) ist nicht Bestandteil des Ingest-Bereichs. Das digitale Archiv sollte sich demnach mit den Datenproduzenten auf ein möglichst strukturell einheitliches SIP verständigen. Mit Blick auf das OAI-Referenzmodell erscheint dieser Vorverarbeitungsabschnitt mangelhaft. Es gibt zwar beschriebene Regelungen dieser Vorarbeiten, aber bislang keine Integration eines

Vorverarbeitungsprozesses im Referenzmodell.<sup>2</sup> Zur Vorbereitung von Informationspaketen als SIP ist eine Erweiterung der Funktionseinheit Pre-Ingest sinnvoll. Der Pre-Ingest-Bereich dient einer spezifischeren Definierung bei der Erstellung des SIPs (Steffenhagen, 2016). Dies kann zur Verbesserung in puncto Datenqualität führen. Durch obligatorische Beschreibungen mittels Metadaten können die Informationspakete im späteren Verlauf besser verwaltet werden. Des Weiteren werden die Strukturen und Formate der Pakete an zentraler Stelle vereinheitlicht. Bei der Entwicklung einer webbasierten Software zur Unterstützung des Vorverarbeitungsprozesses des SIPs gilt es zu unterscheiden, ob ein solches Portal als Pre-Ingest-Werkzeug definiert werden kann. Findet mittels dieser Software lediglich eine Ansammlung und Aufbereitung von Digitalisaten statt, handelt es sich eher um einen Aggregator. Die prototypische Webanwendung, welche im Zuge dieser Bachelorarbeit entwickelt wird, dient in erster Linie der Demonstration als Aggregationsportal. Diese Applikation soll auf modulare Funktionserweiterungen, welche zum Beispiel als Virenüberprüfung oder Validierung von Daten oder Dateien aufwarten, untersucht werden. Die Validation der Metadaten und Dateien wird im Demonstrator teilweise berücksichtigt. Schlussendlich konzentriert sich die Portallösung auf die Aggregation. Allerdings werden Teile eines Pre-Ingest-Werkzeuges abdeckt.

Es ist zudem zu erwähnen, dass die Zuordnung einer Einrichtung durch einen Identifier, dem ISIL, stattfindet.

## **2.5 Der Identifikator ISIL**

Bei einem „International Standard Identifier for Libraries and Related Organizations“ – kurz ISIL – handelt es sich um einen eindeutigen Identifikator beziehungsweise um ein eindeutiges Bibliothekssigel. Das Sigel wird als Abkürzungszeichen einer Einrichtung verwendet, wodurch Bibliotheken in Kurzform gekennzeichnet werden können. Es ist aber auch

---

<sup>2</sup> Beispielsweise bietet der Standard PAIMAS Regelungen von Abläufen und Policies zwischen Produzenten und Archiven bei den Vorarbeiten für das SIP

zur Identifikation anderer verwandter Arten von Einrichtungen, wie Museen oder Archiven, vorgesehen. Auf internationaler Ebene werden seit 2003 solche Sigel in ein ISIL-System eingebunden. Dies geschieht gemäß dem Internationalen Standard ISO 15511 (ISO 15511, 2011). Die Vergabe wird in den meisten Ländern von zentralen Stellen geregelt. In Deutschland gibt es zum Beispiel die Deutsche ISIL-Agentur und Sigelstelle der Staatsbibliothek zu Berlin, welche auch für die Herausgabe von Sigelverzeichnissen verantwortlich ist. Verwendet werden Bibliothekssigel in Verbundkatalogen, im Leihverkehr und Fernleihe sowie zum vereinfachten Datenaustausch. Zukünftige Entwicklungen im Bereich Semantic Web sind nicht auszuschließen. Das Format der ISIL wird in der Norm ISO 15511 geregelt. Ein ISIL besteht aus drei Teilen und kann eine maximale Länge von 16 Zeichen aufweisen. Dieser wird aus einem Präfix, einem Bindestrich und einem Institutionskennzeichen zusammengesetzt. Das Präfix besteht aus maximal vier Großbuchstaben und beinhaltet entweder ein Länderpräfix aus zwei Buchstaben gemäß der ISO 3166-1<sup>3</sup> oder ein Nicht-Länderpräfix aus einem, drei oder vier Buchstaben. Danach folgt ein Bindestrich nach dem ASC II-Code 45 und darauffolgend ein lokales Institutionskennzeichen mit einer maximalen Länge von elf Zeichen. Das Kennzeichen darf aus Groß- und Kleinbuchstaben sowie aus Ziffern und den Sonderzeichen Minus, Schrägstrich und Doppelpunkt bestehen (ISO 15511, 2011).

In der Webanwendung wird der ISIL-Identifikator dem vereinfachten Datenaustausch dienen. Den abzuliefernden Transferpaketen der Einrichtungen wird ein ISIL angehängt, wodurch die Paketverwaltung begünstigt wird. Nach der ISO 15511 kann die Validität eines ISIL-Sigels als Eingabe in einem eingeschränkten Eingabeformular der Webanwendung geprüft werden. Des Weiteren dient der ISIL zur automatischen Erzeugung eines ISIL-Links zum entsprechenden Onlinedatensatz der Staatsbibliothek zu Berlin. Dies wird durch das Hinzufügen des Sigels als GET-Variable in der URL realisiert. Dadurch kann die Angabe der ISIL an geeigneten Stellen als Link dargestellt und der Institutionsdatensatz verknüpft werden.

---

<sup>3</sup> ISO 3166-1, Codes for the representation of names of countries and their subdivisions – Part 1: Country codes

ISIL:  
 -

ISIL-Link (automatisch):\*

\* Der ISIL-Link zur Staatsbibliothek zu Berlin wird automatisch erzeugt, wenn das Feld frei bleibt

Abbildung 2: Auszug des Eingabeformulars zum ISIL während der Produzentenregistrierung im Demonstrator

## 2.6 Die Systemarchitektur des Demonstrators

Die Systemarchitektur eines solchen Archivaggregators orientiert sich zum Teil an das vorher beschriebene OAIS-Funktionsmodell (siehe Kapitel 2.3). Produzent und Endnutzer sind dabei identisch. Als Informationspaket fungiert das Transfer Information Package (TIP). Das TIP wird vom Produzenten hochgeladen und beinhaltet in der Regel eine ZIP-komprimierte Datei mit Bildern im TIFF-Format. Zusätzlich müssen vom Produzenten Metadaten generiert werden. Dies geschieht im Ingest des Portals. Die Metadaten werden in administrative Metadaten (AMD), deskriptive Metadaten (DMD) und technische Metadaten (TMD) unterschieden. Sie werden erfasst und gemeinsam mit einem Statuswert in der Datenbank gespeichert. Die administrativen Metadaten bilden in Kombination mit dem TIP das endgültige Übernahmepaket (TIP\*), welches zunächst im temporären Speicher (STORAGE) abgelegt wird. Das Übernahmepaket kann daraufhin vom Archivsystem importiert und verarbeitet werden. Im digitalen Archiv entsteht folgend das eigentliche SIP für die Funktionseinheit Ingest des Archivsystems. Durchläuft das Paket alle internen Prozesse des Archivsystems, entsteht am Ende das AIP und wird im digitalen Archiv archiviert. Zeitgleich wird eine digitale Kopie an das Portal zurückgeliefert. Aufgrund des Vorhandenseins des AIPs kann das Portal den entsprechenden Bearbeitungsstatus ermitteln und in der Datenbank speichern. Alternativ könnte dieser Status auch vom Archivsystem selbst, durch Zugriff auf die Portaldatenbank, geschrieben werden. Liegt eine digitale Kopie des AIPs vor, kann sie ein Produzent bei Bedarf herunterladen. Damit die Produzenten überhaupt Archivpakete in das Portal hochladen

können, brauchen sie einen Zugang (ACCESS). Dieser wird für alle Nutzergruppen durch HTML arrangiert. Aufgrund dessen können der Datenbankzugriff sowie die Dateiverwaltung stattfinden. Um ein Produzentenkonto zu erhalten, müssen die Einrichtungen von der Administration registriert werden. Eine wichtige Voraussetzung zur Kontoerstellung einer Einrichtung ist die Existenz eines zugewiesenen Sigels bei der Staatsbibliothek zu Berlin. Darüber hinaus kann die Administration für eine Einrichtung agieren. Dies ist aufgrund der hohen Datenmengen erforderlich. Je nach technischen Bedingungen des Servers, auf dem die Webanwendung installiert ist, wird die maximale Dateigröße für hochzuladende Dateien ermittelt und das System dementsprechend begrenzt. Größere Datenbestände müssen manuell, zum Beispiel durch mobile Datenträger, in das System geladen werden (siehe Kapitel 4.4). Für das Webportal bildet der Systemadministrator eine Besonderheit. Er ist verantwortlich für die Instandsetzung des Systems und kann Administratorenkonten bereitstellen. Die Zugriffe des Systemadministrators sind uneingeschränkt. Unter Berücksichtigung der eben genannten Punkte, lässt sich die Systemarchitektur der Webanwendung wie folgt darstellen:

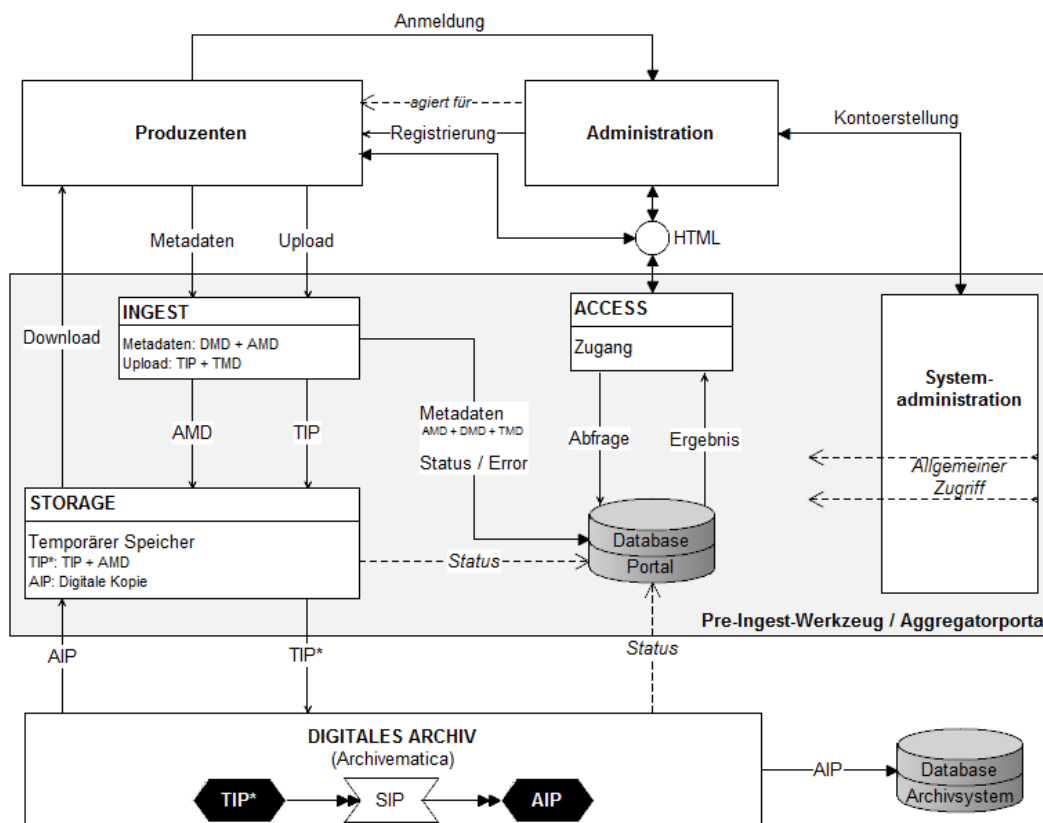


Abbildung 3: Systemarchitektur des Demonstrators

Eine demonstrative Version dieser Webanwendung wird im weiteren Verlauf dieser Arbeit modelliert und entwickelt. Zunächst ist dazu die Datenbankmodellierung essentiell.

## **3 Datenbankmodellierung**

### **3.1 Anforderungen**

Für eine Webanwendung zur Aggregation, Vorverarbeitung und digitaler Archivierung von Digitalisaten sind verschiedene Funktionen und Workflows wesentlich. Auf diese erhalten die drei Nutzergruppen, die auf Grundlage der Systemarchitektur definiert wurden, unterschiedlich beschränkt Zugriff. Dabei erhält der Systemadministrator eine übergeordnete Stellung im Portalsystem, so dass es nicht eines Kontos bedarf, welches mit den Konten der Produzenten und Administratoren kongruent wäre. Sein Konto wird in das Programm integriert. Produzenten und Administratoren hingegen werden in der Datenbank registriert. Produzenten können primär im Namen ihrer Institutionen Daten hochladen, beschreiben, suchen und herunterladen. Die Administration ist primär für die Konten- und Datenverwaltung zuständig. Um ein solches nutzerfähiges System zu entwickeln und um die Datenlieferungen zur Verarbeitung und zur Weiterleitung an das digitale Archiv zu ermöglichen, müssen zunächst alle relevanten Metadaten erhoben werden. Hierzu ist eine möglichst vollständige Identifizierung der denkbaren Datenbasis wichtig.

### **3.2 Identifizierung der Datenbasis**

Unter Berücksichtigung der Anforderungen an die Webanwendung lässt sich das Metadatenpektrum grob in die drei Verwaltungsbereiche, Nutzerverwaltung, Dateiverwaltung und Systemverwaltung, untergliedern. Letzteres wird in der Datenmodellierung des Demonstrators nicht berücksichtigt. Für das prototypische Webportal sind zunächst keine Systemvariablen zur Speicherung in der Datenbank vorgesehen. Im geringen Maß werden solche Variablen in Einzeldateien abgespeichert. Dementsprechend berücksichtigt die Datenbank ausschließlich die Verwaltung der Nutzer und die archivierten Dateien.



Bei der Analyse der Metadaten für die Dateiverwaltung lassen sich die Metadaten zunächst in Kategorien aufteilen. Unter anderem sind die drei mögliche Kategorien, deskriptive, technische und administrative Metadaten, für das Webportal sinnvoll. Die deskriptiven Metadaten dienen hauptsächlich der inhaltlichen und formalen Beschreibung. Mithilfe dieser Beschreibungen können die Daten in einem Retrievalverfahren einfacher zurückgewonnen werden (Qin und Zeng, 2016, S. 19). In der Webanwendung werden die deskriptiven Metadaten für wesentliche Angaben der Archivpakete, deren Datenlieferungen, für Finanzierungsarten und für den Bearbeitungsstatus erhoben. Dabei handelt es sich zumeist um Bezeichnungen und Beschreibungen. Die technischen Metadaten dienen der Charakterisierung technischer Daten von Dateien. Durch sie können theoretisch auch Überprüfungen auf Authentizität und Integrität der Dateien realisiert werden (Qin und Zeng, 2016, S. 19). Derartige Überprüfungen sind für den Pre-Ingest des Demonstrators nicht vorgesehen. Der Demonstrator wird dabei die groben Erkennungsmerkmale, wie Dateityp, Dateigröße und Dateimenge der ZIP-archivierten Datenlieferungen auslesen und speichern. Uhrzeit und Datum der Abgabe einer Datenlieferung kann ebenfalls eine sinnvolle Erweiterung für das Information Retrieval sein. Administrative Metadaten unterstützen die Identifizierung der Objekte, welche in unterschiedlichen Prozessen ineinandergreifen und angesteuert werden müssen. Sie befinden sich in den Daten der zu generierenden Datei „submission\_manifest.txt“, die eine gesamte Lieferung beschreibt (Klindt, Amrhein, Hasler, Oltmanns und Peters-Kottig, 2016, S. 2-3). Die in dieser Datei zusätzlich enthaltenen Metadaten müssen bei der Ermittlung der nötigen Datenbankobjekte berücksichtigt werden und sind teilweise bei der Identifizierung erforderlicher Metainformationen für die Nutzerverwaltung des Systems bedeutend.

Um ein nutzerfähiges Webportal aufzubauen, sollten für die agierenden Akteure Nutzungsrechte und beschreibende Merkmale erfasst und organisiert werden. Angefangen beim Zugang in das Portal lassen sich Zugangsdaten wie ein Passwort, ein dazugehöriger Benutzername und als wesentliche Kontaktquelle eine E-Mail-Adresse ermitteln. Dabei kann auch die E-Mail-Adresse als Benutzername fungieren. Die Kontaktdaten können

um eine Telefonnummer erweitert werden. Für alle Akteure ist wichtig, dass ihnen zugeordnet werden kann, ob sie als Produzent oder Administrator agieren. Aus diesem Grund ist ein Benutzerstatus essentiell. Überdies wird bei den Produzenten erfasst, welcher Organisation sie angehören. Für eine Institution wird zwingend ein ISIL sowie der dazugehörige Link zur Staatsbibliothek zu Berlin vermerkt. Des Weiteren sollte der Name der Institution und ein allgemeiner und administrativer Ansprechpartner mit einer E-Mail-Adresse erfasst werden. Im nächsten Schritt kann mit der Ansammlung der nun relevanten Metadaten für die Datenmodellierung eine spezifische Identifizierung der einzelnen Datenbankobjekte und Attribute erfolgen.

Innerhalb der Attribut- und Objektidentifizierung der Datenbank werden die Objekttypen und Objekteigenschaften auf Grundlage der vorangegangenen Überlegungen in der Metadatenanalyse identifiziert und beschrieben. Wichtig in diesem Prozess ist, dass die funktionalen und transitiven Abhängigkeiten zwischen den Relationen möglichst einwandfrei sind. Überprüfungen der sogenannten Normalformen werden an späterer Stelle näher erläutert (siehe Kapitel 3.5). Durch die Normalisierung wird die Anwendung der Datenbank erleichtert und entstandene Redundanzen können reduziert werden. Das Datenbankschema soll sich durch eine gute Übersichtlichkeit auszeichnen (Faeskorn-Woyke, Bertelsmeier, Riemer und Bauer, 2007, S. 155-156). Untersucht und ermittelt werden die Entitäten (Tabellen), Attribute (Tabellenspalten) und die Schlüssel des Datenbankmodells. Die Entität bildet ein reales Objekt, Begriffe oder Personen ab, die in der Datenbank beschrieben und erfasst werden. Charakterisiert werden die Tabellen wiederum durch ihre Tabellenspalten und durch deren Werte. Attribute sind damit die Eigenschaften der Objekte. Durch die Zuordnung eines individuellen Schlüssels kann eine Entität eindeutig identifiziert werden. Im Vergleich zu den Attributen, welche optional oder obligatorisch sein können, sind Schlüssel immer verpflichtend (Faeskorn-Woyke et al., 2007, S. 83-86). Bei der Festlegung von Tabellenspalten müssen diverse Problemquellen berücksichtigt werden. Es sollen nur Spalten definiert werden, die für die Problemstellung Relevanz besitzen. Die Bezeichnungen sollten innerhalb

einer Dateneinheit eindeutig vergeben werden. Des Weiteren sind Umlaute und Sonderzeichen, abgesehen vom Unterstrich, sowie Zahlen am Anfang der Namensvergabe zu vermeiden (Faeskorn-Woyke et al., 2007, S. 89). Identifizieren und zusammenfassen lassen sich alle Objekttypen und Objekteigenschaften in tabellarischer Form (siehe Anhang A). Nach der tabellierten Zusammenstellung ermittelter Attribute lassen sich zunächst vier Haupttabellen (Akteur, Institution, Archivpaket und Datenlieferung) sowie drei Nebentabellen (Benutzerstatus, Bearbeitungsstatus und Finanzierungsart) feststellen. Demzufolge besteht das Datenbankmodell aus sieben Entitäten. Unter Umständen werden noch weitere Hilfstabellen benötigt. Um dies zu konstatieren, werden nachfolgend die Objektbeziehungen zwischen allen Dateneinheiten untersucht.

### **3.3 Untersuchung der Objektbeziehungen**

Während dieser Untersuchung werden alle sieben identifizierten Objekttypen paarweise gegenübergestellt, um die Beziehungen zwischen den einzelnen Entitäten zu bestimmen. Die Relationen beschreiben einen bestehenden Zusammenhang zwischen den Entitäten und ihren möglichen Mengen (Daten). Sie können ebenfalls zusätzliche Attribute enthalten. Die Kohärenz wird in der Regel durch ein Verb beschrieben (Faeskorn-Woyke et al., 2007, S. 84). Geprüft werden nun die Arten der Relation. Für dieses Datenmodell werden drei Arten unterschieden: die 1:1-Beziehung, die 1:n-Beziehung und die n:m-Beziehung. Bei der 1:1-Beziehung handelt es sich um einen Ausnahmefall. Dabei gehört zu einer Entitätsmenge genau eine Menge einer anderen Entität. Zumeist lassen sich die Einzelmengen zusammenfassen (Faeskorn-Woyke et al., 2007, S. 89-90). Trotzdem findet diese Relation in diesem Datenbankmodell bei der Verknüpfung von Akteuren und Institutionen Anwendung. Solange es vorgesehen ist, dass eine Institution nur durch ein Konto eines einzigen Akteurs (Produzent) im Portal agieren kann, ist eine Segmentierung nicht zwingend notwendig. Dennoch – die Aufteilung wurde zugunsten möglicher Vorteile umgesetzt. Zum einen die Vereinfachung zukünftiger Erweiterungen, dass die Daten einer Einrichtung

von mehreren Produzenten verwaltet werden können.<sup>4</sup> Zum anderen die Steigerung der Systemperformance bei Datenbankabfragen und Datenbankaktionen. Außerdem erhöht es die Übersichtlichkeit des Datenbankmodells. Des Weiteren existiert zwischen dem Akteur und der Institution eine nicht permanente Kann-Beziehung, da die Administratoren keiner erfassten Einrichtung angehören. Die 1:n-Beziehung zeichnet sich dadurch aus, dass zu einer Entitätsmenge mehrere Mengen einer Entität gehören können. Bei einer n:m-Beziehung können demnach mehrere Entitätsmengen mit den Mengen einer anderen Entität verknüpft werden (Faeskorn-Woyke et al., 2007, S. 91-92). Während der Untersuchung der Objektbeziehungen werden auch die Wertebereiche möglicher Mengen in Minimal- und Maximalwerten festgelegt und infolgedessen die Optionalitäten berücksichtigt. Die paarweisen Relationen zwischen den sieben Tabellen können wie im Anhang dargestellt sinnvoll beschrieben werden (siehe Anhang B). Aus den dortigen acht festgelegten Objektbeziehungen und den zuvor identifizierten Attributen lässt sich im folgenden Schritt das Entity-Relationship-Modell ableiten.

### **3.4 Das Entity-Relationship-Modell (ERM)**

Das Entity-Relationship-Modell ist eine Möglichkeit zur Erstellung eines konzeptionellen Entwurfs für das spätere Datenbankschema. Die Entitäten, Attribute und Relationen und die definierten Wertebereiche werden in diesem Modell kompakt in grafischer Form zusammengetragen. Die Visualisierung ist das sogenannte ER-Diagramm, dessen Notationsformen unterschiedlich dargestellt werden können.<sup>5</sup> Vorrangig dient ein ER-Modell als Medium zur Kommunikation zwischen den Datenbankentwicklern und den Anwendern (Faeskorn-Woyke et al., 2007, S. 74-77). Für den Produzenten als Endnutzer hat das ER-Modell keine Relevanz. In diesem Schritt wird die Abbildung eines ER-Diagramms (siehe Anhang C) mit den möglichst genau vordefinierten Datenbanksachverhalten zur Problematik entwickelt. Den Entitäten (Rechtecke) werden die entsprechenden Attribute (Textboxen) zugeordnet. Dabei wird jeweils der Primärschlüssel mit einem

---

<sup>4</sup> Dadurch würde die 1:1-Beziehung in eine 1:n-Beziehung umgewandelt.

<sup>5</sup> Die Chen-Notation nach Peter Chen ist ein Beispiel für eine grafische Darstellung eines ER-Modells.

voranstehenden Doppelkreuz gekennzeichnet. Die Objektbeziehungen bleiben laut vorangegangener Untersuchung bestehen. Zwischen den Entitäten Archivpaket und Finanzierungsart ist auffällig, dass eine Hilfstabelle erforderlich ist. Auf diesen Sachverhalt wird im nächsten Abschnitt eingegangen. Dort wird das bisherige ER-Modell auf diverse Kriterien hin überprüft.

### 3.5 Überprüfung des ER-Modells

Das bisherige konzeptionelle Schema kann auf diverse Eigenschaften hin überprüft werden. Dabei werden zugleich die Normalformen gebildet. Die Prüfung erfolgt nach der funktionalen und transitiven Abhängigkeit, nach Schlüsseln und der Datenintegrität. Durch den Normalisierungsprozess soll das Datenmodell möglichst verständlich, pflegeleicht und redundanzfrei werden. Unvereinbare Abhängigkeiten in der Datenstruktur und eventuelle Regelwidrigkeiten durch Datenbankaktionen<sup>6</sup> werden eliminiert. Insgesamt soll dadurch die Nachhaltigkeit des Datenbankschemas positiv beeinflusst werden (Faeskorn-Woyke et al., 2007, S. 149-155). Das Schema wird zunächst auf drei Normalformen überprüft. Die erste Normalform besteht, wenn die Mengen einzelner Attribute atomarisiert sind. Zusammengesetzte Datentypen müssen also gegebenenfalls in mehrere Attribute geteilt werden (Faeskorn-Woyke et al., 2007, S. 156-157). Ein Beispiel dafür wäre der vollständige Personennamen in einer Tabellenspalte „Name“, welcher aus Vor- und Zunamen besteht und demzufolge in zwei Tabellenspalten „Vorname“ und „Nachname“ geteilt werden kann. In den Entitäten Institution und Archivpaket können Ansprechpartner und Kuratoren erfasst werden. Für diese sind Segmentierungen in weitere Tabellen nicht vorgesehen. Über das Webportal kann die Person in einem Eingabefeld beliebig eingegeben und gespeichert werden, sofern das Portal die Eingabestruktur nicht kongruent zum Datenbankmodell festlegt. Dies ist nicht optimal, wenn das Portal explizite Suchoperationen in diesen Datenfeldern bereitstellt. Da solche Felder ebenfalls nicht vorgesehen sind, sondern nur eine Datensatzausgabe

---

<sup>6</sup> INSERT (einfügen), UPDATE (ändern) und DELETE (löschen)

als Übersicht einzelner Transferpakete oder Projekte, hat eine Aufteilung an dieser Stelle keine Bewandnis. Aufteilungen von Attributen des bisherigen ER-Modells sind nicht zwingend erforderlich, da keine störenden Datentypen, die zusammengesetzt oder mehrwertig sind, bestehen. Im weiteren Verlauf kann auf die zweite Normalform geprüft und die Schlüssel- und Nicht-Schlüsselattribute auf Abhängigkeiten ihrer Werte untersucht werden. Der Primärschlüssel bildet in jeder Relation einen eindeutigen Schlüssel, durch welchen die Datensätze eindeutig identifiziert werden können (Faeskorn-Woyke et al., 2007, S. 152-160). Alle Nicht-Schlüsselattribute einer Entität sollen lediglich vom Primärschlüssel funktional abhängig sein. Ein ISIL, als eindeutiger Primärschlüssel der Entität „Institution“, übt beispielhaft eine funktionale Abhängigkeit auf alle Nicht-Schlüsselattribute derselben Tabelle aus. Mehrfache Datenspeicherungen tauchen in den Dateneinheiten nicht auf. Es müssen keine weiteren Entitäten gebildet werden, um die zweite Normalform zu erreichen. Die Überprüfung der dritten Normalform ist ein vergleichbares Verfahren zur zweiten Normalform. Hierbei werden die transitiven Abhängigkeiten der Nicht-Schlüsselattribute paarweise untersucht. Die Nicht-Schlüsselattribute einer Relation dürfen dabei nicht voneinander abhängig sein (Faeskorn-Woyke et al., 2007, S. 160-163). Wird zum Beispiel in der Tabelle „Akteur“ die E-Mail-Adresse geändert, so kann das gleiche Passwort bestehen bleiben, wodurch keine Abhängigkeit vorliegt. Paarweise müssen so alle möglichen Attributkombinationen innerhalb einer jeden Entität überprüft werden. Gegebenenfalls können auf diese Weise neue Entitäten entstehen, was im Falle des Webportals nicht notwendig ist. Das Normalisierungsverfahren sowie die Datenintegrität sollten bestenfalls während des gesamten Entwicklungsprozesses berücksichtigt werden. Widersprüche und Redundanzen der Daten sollen verhindert werden. Die Konsistenz und folgend die Korrektheit der zu verwaltenden Daten kann durch Relationen und vordefinierten Attributdatentypen beschränkt werden. Ein Beispiel für eine weitere Relation ist die zusätzliche Tabelle „Benutzerstatus“, welche die Eintragung des korrekten Benutzerstatuswertes gewährleistet. Vor der Überführung des ER-Modells in ein relationales Datenbankmodell können Normalisierungsprozesse stattfinden. Das

Datenbankschema, das im nächsten Schritt erzeugt wird, soll hingegen normalisiert werden.

### 3.6 Transformation von ERM in RDM

Das relationale Datenbankmodell (RDM) soll auf Grundlage des fertigen konzeptionellen ER-Modells abgeleitet werden. Das RDM ist als relationales Datenbankschema auf einen relationalen Datenbanktyp abgestimmt (Faeskorn-Woyke et al., 2007, S. 75). Während der Transformation werden die Entitäten auf Relationen und diese wiederum auf Fremdschlüsselattribute abgebildet (Faeskorn-Woyke et al., 2007, S. 165). Die Ausgangsform, ausgehend vom ER-Modell, sieht wie folgt aus:

---

S {Akteur, Institution, Archivpaket, Datenlieferung, Benutzerstatus, Bearbeitungsstatus, Finanzierungsart}

R<sub>Akteur</sub>(#akteur\_id, benutzername, passwort, email, telefonnummer)

R<sub>Institution</sub>(#ISIL, ISIL\_link, institutionsname, ansprechpartner, ansprechpartner\_email)

R<sub>Archivpaket</sub>(#archivpaket\_id, projekttitle, projektbeschreibung, jahr, kurator, kurator\_email, rechte, paketverzeichnis)

R<sub>Datenlieferung</sub>(#datenlieferung\_id, dateiname, dateiname\_intern, dateityp, ablieferungsbeschreibung, ablieferungsdatum, ablieferungszeit, dateigroesse, dateimenge)

R<sub>Benutzerstatus</sub>(#benutzerstatus\_id, bezeichnung)

R<sub>Bearbeitungsstatus</sub>(#bearbeitungsstatus\_id, bezeichnung, beschreibung)

R<sub>Finanzierungsart</sub>(#finanzierungsart\_id, art)

---

Zwischen den Entitäten „Archivpaket“ und „Finanzierungsart“ ist eine Hilfstabelle notwendig. Dies erschließt sich aus dem Beziehungstyp, der n:m-Beziehung, die auf eine Mehrfachauswahl von Finanzierungsarten zurückzuführen ist. Aufgrund dessen entsteht eine neue Relation „Finanzierung“. Die Datenstruktur wird demnach um eine weitere Tabelle ergänzt. Mittels Fremdschlüssel werden nun die Dateneinheiten durch ihre Attribute kombiniert. Die Fremdschlüssel sind dabei Primärschlüssel einer anderen Tabelle, welche in den zu verknüpfenden Relationen eingefügt werden (Faeskorn-Woyke et al., 2007, S. 152). In welche Entität der Primärschlüssel

einer Tabelle als Fremdschlüssel einer anderen Tabelle eingefügt wird, ist abhängig vom Beziehungstyp und dessen Bestimmung. Wird die Ausgangsform des Datenbankschemas um die entstehende Hilfstabelle und den Fremdschlüsseln ergänzt, entsteht folgende Datenbankstruktur:

---

S {Akteur, Institution, Archivpaket, Datenlieferung, Benutzerstatus, Bearbeitungsstatus, Finanzierungsart, Finanzierung}

R<sub>Akteur</sub>(#akteur\_id, benutzername, passwort, email, telefonnummer, ISIL, benutzerstatus\_id)

R<sub>Institution</sub>(#ISIL, ISIL\_link, institutionsname, ansprechpartner, ansprechpartner\_email)

R<sub>Archivpaket</sub>(#archivpaket\_id, projekttitle, projektbeschreibung, jahr, kurator, kurator\_email, rechte, paketverzeichnis, akteur\_id, ISIL)

R<sub>Datenlieferung</sub>(#datenlieferung\_id, dateiname, dateiname\_intern, dateityp, ablieferungsbeschreibung, ablieferungsdatum, ablieferungszeit, dateigroesse, dateimenge, akteur\_id, archivpaket\_id, bearbeitungsstatus\_id)

R<sub>Benutzerstatus</sub>(#benutzerstatus\_id, bezeichnung)

R<sub>Bearbeitungsstatus</sub>(#bearbeitungsstatus\_id, bezeichnung, beschreibung)

R<sub>Finanzierungsart</sub>(#finanzierungsart\_id, art)

R<sub>Finanzierung</sub>(finanzierungsart\_id, archivpaket\_id)

---

Diese Ansicht ist zugleich das endgültige relationale Datenbankschema. Bestand das ursprüngliche konzeptionelle Schema noch aus sieben Tabellen, besitzt das relationale Datenmodell nun acht Tabellen. Nachkommend kann mithilfe der grafischen Repräsentation des entstandenen RDM (siehe Anhang D) die prototypische Webanwendung konzeptioniert und entwickelt werden.



## **4 Entwicklung der prototypischen Webanwendung**

### **4.1 Allgemeines und Methodik**

Im Verlauf dieser Bachelorarbeit wurde eine prototypische Webanwendung zur Aggregation der Digitalisate entwickelt. Es handelt sich dabei um ein Demonstrationsportal, welches als Aggregator fungiert, eine Nutzerverwaltung beinhaltet und auf Erweiterungen zu einem Pre-Ingest-Werkzeug analysiert werden soll. Der webbasierte Demonstrator wurde mittels Auszeichnungs-, Programmier- und Datenbanksprachen umgesetzt. Zur Strukturierung der Webseiteninhalte wird die Auszeichnungssprache HTML verwendet, die unter Anwendung der Stylesheet-Sprache CSS gestaltet wird. Als weitere Auszeichnungssprache kommt XML zum Einsatz, um die aggregierten Metadaten zunächst in einer einfachen Struktur zu speichern. Die Programmiersprache PHP dient der serverseitigen Programmierung und wurde neben dem Funktionsaufbau dazu verwendet einerseits die MySQL-Datenbank zu installieren und mit dieser zu kommunizieren und andererseits um die Webseitenstruktur in Kombination mit HTML zu bilden. In PHP wird zur Datenbankkommunikation die Datenbanksprache SQL angewandt. Im geringen Maß wurde für nutzerseitige Funktionen über den Browser die objektorientierte Programmiersprache Java eingesetzt. Das demonstrative Portal soll als Basis für Analysen fungieren. Es soll beurteilt werden inwieweit die Webanwendung die definierten Anforderungen erfüllt und ob es als Grundlage weiterer Entwicklungsprozesse geeignet ist. Außerdem soll ermittelt werden, welche Fehler bei der Umsetzung auftauchen und welche Probleme durch ein solches Portal entstehen können. Aufgrund dessen können mögliche Verbesserungen oder alternative Ansätze gesucht werden. In diesem Abschnitt soll die Entwicklung erläutert werden.

## 4.2 Verzeichnisstruktur

Vor der programmtechnischen Umsetzung der Webanwendung wurde eine hierarchische Verzeichnisstruktur festgelegt (siehe Anhang E). Die Struktur beinhaltet vier Hierarchieebenen. Die erste Ebene besteht aus dem Skript „index.php“, welches als Startseite beziehungsweise als Standard-Index-Datei vom Webserver automatisch ausgeführt wird, sobald ein Nutzer die Webseite im Browser aufruft. Hinzu kommen auf gleicher Ebene alle relevanten Verzeichnisse, beispielsweise für Unterseiten, Funktionen, Bilder und Dateien, sowie die Dateien „.htaccess“ für Konfigurationen<sup>7</sup> des Servers und „robots.txt“ für Anweisungen an Webcrawler, welche die Webseiten durchsuchen und analysieren. Neben den eigentlichen Komponenten der Webanwendung sind beim Aufbau der Verzeichnisstruktur zwei verwaltungstechnische Aufgaben zu berücksichtigen: die Dateiverwaltung der Transferpakete und die Verwaltung der Metadaten in Dateiformaten. Für die Transferpakete der Produzenten existiert das Verzeichnis „archiv“. Dort werden hierarchisch die Dateien abgelegt. Bei der Verzeichnisstruktur für Transferpakete gilt es einen Kompromiss herzustellen, „zwischen größtmöglicher Flexibilität für den Datengeber und Minimierung des Aufwands für das digitale Archiv“ (Klindt et al., 2016, S. 5). Im Verzeichnis „archiv“ wird je Produzent ein Verzeichnis auf zweiter Hierarchieebene angelegt. Der Verzeichnisname besteht dabei aus „produzent\_“ und der Datenbank ID des Produzenten. In jedem Produzentenverzeichnis können beliebig viele Ordner angelegt werden. In diesem Zusammenhang repräsentiert jedes Verzeichnis ein einzelnes Projekt, das vom Produzenten mithilfe des Aggregators erstellt wird. Die Bezeichnung des Projektordners folgt dabei keinem Schema und entspricht dem vergebenen Projekttitel. In jedem Projektverzeichnis befinden sich wiederum beliebig viele Datenlieferungen im ZIP-komprimierten Format und zwei Metadatendateien. Eine Datenlieferung trägt eine Bezeichnung, die sich aus „transfer\_“ und der ID-Nummer der Datenlieferung zusammensetzt. Für Metadaten-Dateien sind für jedes Projekt jeweils eine „metadaten.xml“ und eine

---

<sup>7</sup> Sofern der Server es zulässt Systemvariablen der Datei „php.ini“ zu modifizieren, ändert das Programm durch die Datei „.htaccess“ definierte Variablen (z. B. die maximale Dateigröße beim Upload)

„submission\_manifest.txt“ intendiert. Metadaten und Digitalisate bilden gemeinsam eine intellektuelle Einheit (IE). Diese mögliche Variante fasst Einzeldateien einer IE in jeweilige Unterverzeichnisse zusammen (Klindt et al., 2016, S. 5-6). Die Verzeichnisstruktur für Transferpakete kann folgendermaßen dargestellt werden:

```

archiv/
|----- produzent_1/
|         |----- projekt_001/
|         |         |----- transfer_1.zip
|         |         |----- transfer_2.zip
|         |         |----- transfer_3.zip
|         |         |----- metadaten.xml
|         |         |----- submission_manifest.txt
|         |----- projekt_002/
|         |         |----- transfer_4.zip
|         |         |----- metadaten.xml
|         |         |----- submission_manifest.txt
|----- produzent_2/
|         |----- projekt_003/
|         |         |----- transfer_5.zip
|         |         |----- transfer_6.zip
|         |         |----- metadaten.xml
|         |         |----- submission_manifest.txt

```

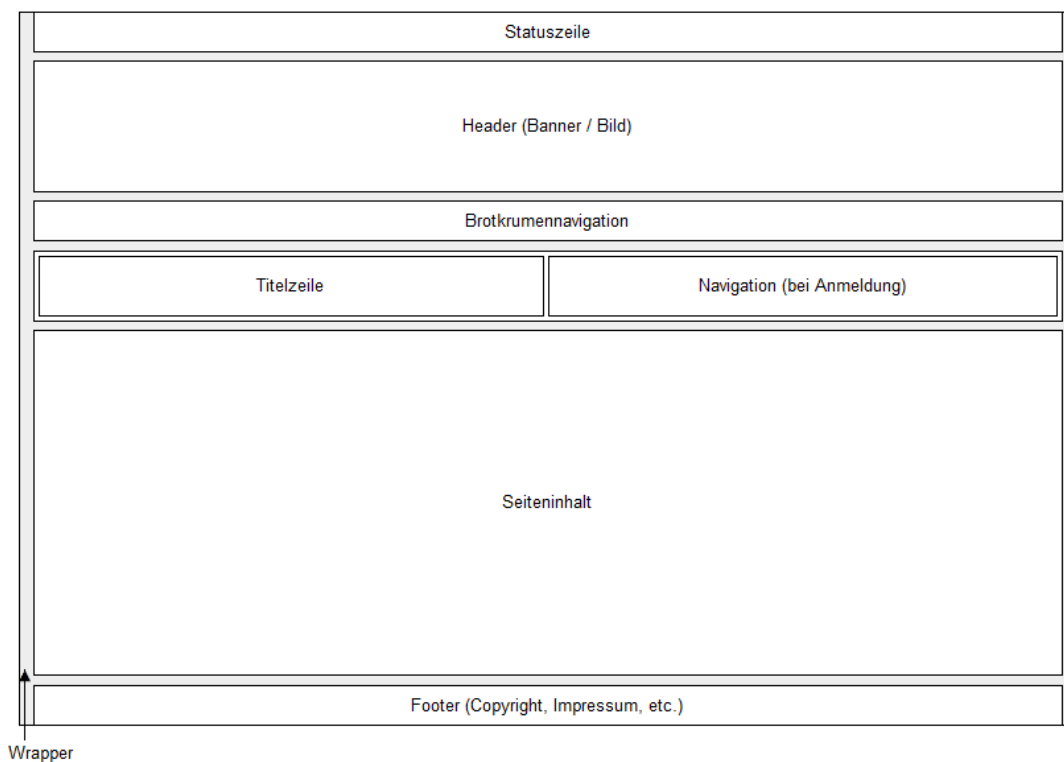
**Abbildung 4: Verzeichnisstruktur zur Speicherung von Transferpaketen im Demonstrator**

Die ZIP-komprimierten Transferpakete werden als TIP in das jeweilige Projektverzeichnis abgelegt. Nachdem die Pakete alle Verarbeitungs- und Archivierungsprozesse des Archivsystems durchlaufen haben, werden sie als digitale Kopie des AIPs im selbigen Verzeichnis abgelegt und können vom Produzenten heruntergeladen werden. Damit dieser Prozess stattfinden kann, muss die jeweilige Datenlieferung in den temporären Speicher kopiert werden. Dieser befindet sich in der Verzeichnisstruktur im Ordner „temporaer“ und wird untergliedert in Input und Output. Der Transferprozess wird durch einen Administrator gestartet. Dabei wird das Transferpaket in das Input-Verzeichnis kopiert und alle enthaltenen Dateien entpackt. Zusätzlich werden die beiden Metadatendateien mitgeliefert. Das Archivsystem kann nun das Input-Verzeichnis durchsuchen, analysieren und die Transferpakete entnehmen. Das TIP, bestehend aus den Datenlieferungen und den obligatorischen administrativen Metadaten, gelangt über diesen Weg als SIP

in das Archivsystem. Das AIP wird, nachdem es im digitalen Archiv gespeichert wird, als digitale Kopie in das Output-Verzeichnis zurückgeliefert. Das Format sollte dabei erneut ZIP-komprimiert sein. Diesmal wird der Aggregator das Output-Verzeichnis durchsuchen und analysieren. Wenn ein AIP vorhanden ist, kann es aufgrund der gespeicherten Statuswerte in der Datenbank dem entsprechenden Projekt zugeordnet und in das ursprüngliche Produzentenverzeichnis verschoben werden. Der temporäre Speicher (Input und Output) wird nach diesem Prozess wieder geleert. Der gesamte Prozess der Archivsystemkommunikation findet in der Webanwendung nur demonstrativ als Simulation statt (siehe Kapitel 4.5). Nach der Planung und Festlegung einer Verzeichnisstruktur für die Webanwendung und einer Dateiverwaltung für die Transferpakete fand im nächsten Schritt der grundstrukturelle Aufbau der Webseiten statt.

### **4.3 Grundstruktur der Webseiten**

Die webbasierte Anwendung als Demonstrator bedarf kaum Entwicklungen für das Design. Ein rudimentäres Layout der Webseiten als konzeptionellen Entwurf ist hingegen empfehlenswert. Eine schematische Darstellung dazu bietet der sogenannte Wireframe. In diesem werden die Elemente einer Webseite mittels geometrischer Formen modellhaft gekennzeichnet. Das eigentliche Design ist in diesem Prozess noch unwesentlich. Für den Demonstrator wurde ein einfacher, statischer Wireframe für die Startseite entworfen:



**Abbildung 5: Wireframe des Demonstrators**

Das Layout ist zugleich Vorlage aller weiteren Unterseiten. Die entwickelten Skripte der Webseiten haben alle den gleichen Aufbau. Sie bestehen aus einer Kombination der Sprachen HTML und PHP. Das HTML bezieht sich dabei auf die Empfehlungen der W3C-Spezifikation zu HTML5.<sup>8</sup> Durch die Kombination können so seitenübergreifende Elemente zentral angepasst und in jedes Skript geladen werden – ähnlich eines Content Management Systems (CMS). Dies betrifft alle im Wireframe enthaltenen Elemente mit Ausnahme des eigentlichen Seiteninhalts. Des Weiteren werden durch PHP weitere PHP-Skripte und -Funktionen, CSS- und Java-Einbindungen sowie HTML-Metainformationen im Head-Element in jedes Skript eingebunden. Prinzipiell ist je nach Erfordernissen ein Skript in etwa 17 Teilen wie folgt aufgebaut:

1. Allgemeine PHP Konfigurationen
2. Vordefinierte PHP Variablen
3. Laden vorgefertigter Skripte und Funktionen

<sup>8</sup> Hickson, Ian et al. (2014). HTML5. A vocabulary and associated APIs for HTML and XHTML. W3C Recommendation. Zugriff am: 15.02.2017, verfügbar unter: <https://www.w3.org/TR/html5/>

4. Überprüfung der Zugangsberechtigung des Nutzers
5. Programm des Skriptes (Programmausführung)
6. Eröffnung des HTML-Dokuments
7. HTML-Head-Element: Titel und Metainformationen
8. Eröffnung des HTML-Body-Elements
9. Inhaltsverzeichnis im Skript als Kommentar
10. Statuszeile
11. Header
12. Brotkrumennavigation
13. Titelzeile und Navigation
14. Seiteninhalt
15. Footer
16. Schließen des HTML-Body-Elements
17. Schließen des HTML-Dokuments

Zunächst sind die Punkte eins bis vier interessant. Anfänglich werden diverse Konfigurationen durchgeführt. In der Regel handelt es sich hierbei um das Ausblenden von serverseitigen PHP-NOTICE Hinweisen. Außerdem wird eine gegebenenfalls bestehende Session fortgeführt. Durch die Session beziehungsweise einer Sitzung werden Variablen über den Nutzer skriptübergreifend gespeichert und im Browser des Nutzers temporär hinterlegt. Danach werden wiederkehrende Ausführungen von PHP-Codezeilen in PHP-Skripte im Verzeichnis „php“ ausgelagert. Die dort ebenfalls enthaltene Datei „skriptbibliothek.php“ fasst alle Einzelskripte zusammen und wird in jede Webseite eingebunden. Ein Beispiel hierfür sind die Zugangsdaten für die MySQL-Datenbank, die in Variablen abgespeichert werden. PHP-Funktionen verhalten sich nach dem gleichen Prinzip. Diese werden im Verzeichnis „funktionen“ durch das Skript „funktionsbibliothek.php“ zusammengefasst und durch dieses eingebunden. Dadurch kann PHP jederzeit auf Funktionen – wie zum Beispiel eine Datenbankabfrage – zugreifen. Punkt vier beschreibt die Überprüfung beim Seitenaufruf durch PHP, ob der jeweilige Nutzer berechtigt ist, den Inhalt zu sehen. Andernfalls wird dem Nutzer der Zugriff verwehrt. Nach den genannten Punkten folgen

die eigentliche Programmausführung eines jeden Skriptes und die Ausgabe im HTML-Format.

#### **4.4 Programmierung der Webanwendung**

Alle benötigten Grundvoraussetzungen zur funktionsfähigen Umsetzung der Webanwendung sind gegeben. Dazu zählen die Grundstrukturen der Webseiten in Kombination der Sprachen HTML und PHP, in denen wichtige Skripteinbindungen durch die PHP-Funktionen „include“ oder „require“ (bei obligatorischen Skripten) festgelegt wurden. Des Weiteren dienen das relationale Datenbankmodell und die Systemarchitektur als Grundlage. Zunächst wurde mit HTML, PHP und CSS die Grundstruktur in einem einfachen Design, durch Erstellung aller erforderlichen Boxen und Formatierungen sämtlicher Elemente, umgesetzt. In CSS wird dabei eine Variable verwendet, mit welcher sich die Farbgebung der gesamten Webseiten konfigurieren lässt. Eine Besonderheit der Grundstruktur ist die Navigation. Diese ist nur für angemeldete Nutzer sichtbar. Konstante Links befinden sich im Bereich der Statuszeile und des Footers. Der Footer beinhaltet das Impressum und eventuell weitere Webseiten wie zum Beispiel Kontakt oder ein FAQ (Frequently Asked Questions). Über die Statuszeile erreicht der Nutzer die Anmeldung. Die Skripte zum An- und Abmeldeverfahren befinden sich im Verzeichnis „zugang“. Das Login verlangt ein Passwort sowie einen Benutzernamen der eine E-Mail-Adresse, ein ISIL oder ein festgelegter Benutzername sein kann. Wenn das Passwort vergessen wurde, kann ein neues Passwort angefordert werden, welches per E-Mail mitsamt eines Aktivierungslinks gesendet wird. Diese E-Mail wird mangels Webserver im Testsystem lediglich simuliert. Die Installation der Datenbank, die die Nutzerkonten zur Anmeldung enthält, kann nur vom Systemadministrator durchgeführt werden. Dieser erhält ein datenbankunabhängiges Konto. Der Benutzername „Admin“ wird für dieses Konto reserviert, wobei das Passwort in einer externen Textdatei im Verzeichnis „zugang“ gespeichert ist und geändert werden kann. Die

Navigation beinhaltet fünf Hauptkategorien. Der Zugriff auf eine Kategorie ist benutzerabhängig.

Kategorie\Nutzergruppe	Produzent	Administrator	Systemadministrator
System	-	-	+
Verwaltung	-	+	+
Profil	+	+	+
Suche/Herunterladen	+	+	+
Hochladen	+	+	+

Tabelle 1: Zugriffsrechte auf Programmabschnitte nach Nutzergruppen

Bestimmten Nutzergruppen werden fragmentarisch Skripte innerhalb einer Kategorie vorbehalten. Im Folgenden werden die Aufgaben der fünf Kategorien als Programmabschnitte näher erläutert.

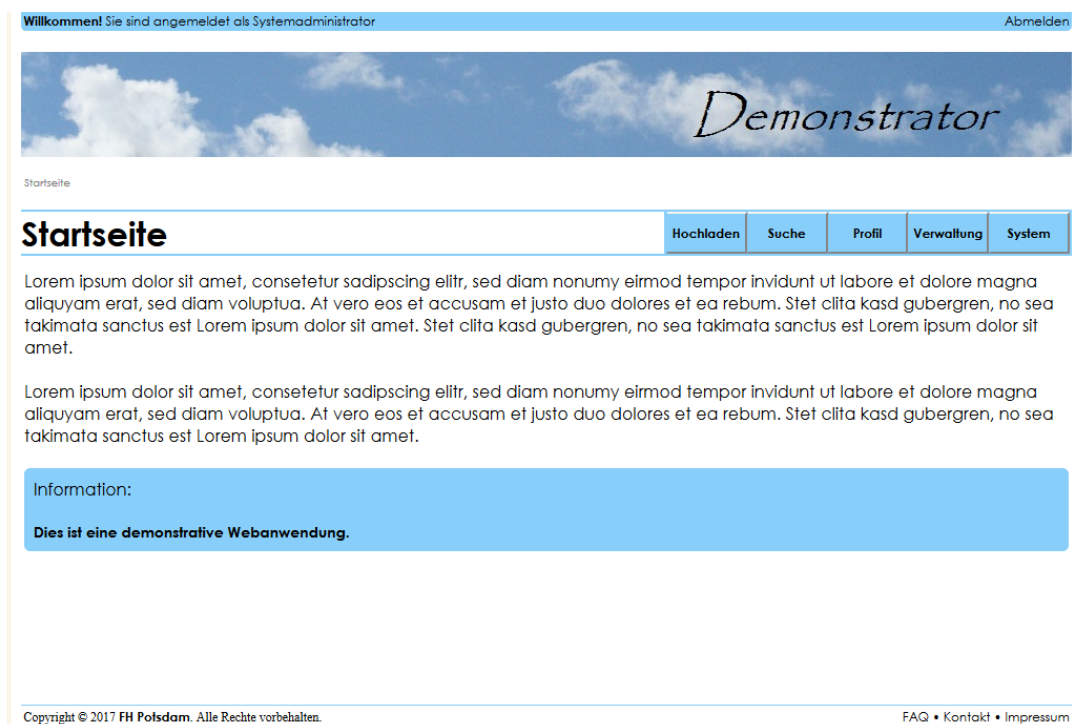


Abbildung 6: Screenshot der Startseite des Demonstrators nach der Anmeldung des Systemadministrators

### 1. Programmabschnitt: System

Der Zugriff auf den Programmabschnitt „System“ ist nur einem Systemadministrator gestattet. Skripte dieses Abschnittes werden im Verzeichnis „system“ abgelegt. Die wesentliche Aufgabe des Bereiches ist die Installation oder Deinstallation der MySQL-Datenbank. Das Programm überprüft die Verbindung zur Datenbank und den Status. Anhand eines



Status kann der Systemadministrator durch einen Schalter die Installation oder Deinstallation veranlassen. Bei der Installation der Datenbanktabellen wurde die Engine MyISAM gewählt. Da dieses Speichersubsystem keine Indexdaten auslesen muss, begünstigt sie einen schnelleren Zugriff auf Tabellen. Vor allem ermöglicht sie die Erstellung eines Volltextindexes und damit eine Volltextsuche in der Beschreibungen der einzelnen Projekte und deren Datenlieferungen enthalten sind. Sie unterstützt jedoch keine Fremdschlüssel-Transaktionen im Gegensatz zur standardmäßigen Engine von MySQL InnoDB, wobei auch diese Einschränkungen unterliegt (Faeskorn-Woyke et al., 2007, S. 488-490). Aufgrund dessen wurde zugunsten einer Volltextsuche die Engine MyISAM gewählt und die Transaktionen über Fremdschlüssel durch SQL im PHP-Code berücksichtigt.

## 2. Programmabschnitt: Verwaltung

Auch der Abschnitt „Verwaltung“ ist nur den (System-)Administratoren erlaubt. Skripte dieses Abschnittes werden im Verzeichnis „verwaltung“ abgelegt. Die Verwaltungsaufgaben lassen sich in drei Bereiche gliedern: die Verwaltung von Webseiten des Aggregators, die Datenbankverwaltung und die Nutzerverwaltung. Die Webseiten Startseite, Impressum, Kontakt und FAQ sind inhaltlich in Einzelskripten anpassbar. Die Informationsbox der Startseite ist zudem ein- und ausblendbar. Gespeichert werden die Daten in Textdokumenten. In der Datenbank können die Finanzierungsarten durch hinzufügen und löschen verwaltet werden. Konten für Produzenten und Administratoren können registriert und übersichtlich angezeigt werden (siehe Anhang F). Die Passwörter werden während der Registrierung mit dem MD5-Verfahren verschlüsselt.

## 3. Programmabschnitt: Profil

Im Abschnitt „Profil“ können alle Nutzer ihre Kontodaten einsehen. Außerdem wird hier die Möglichkeit bereitgestellt das Passwort zu ändern. Die Skripte des Abschnittes liegen im Verzeichnis „profil“. Ein Skript zu Änderungen weiterer Profildaten wurde nicht umgesetzt. Derartige Änderungen können zu Problemen bezüglich bereits vom Archivsystem übernommene

Metadateien führen und somit Identifizierungen und Zuordnungen von Einrichtungen und Datenlieferungen beeinträchtigen.

#### 4. Programmabschnitt: Suche / Herunterladen

Je nach Nutzergruppe ändert sich die Bezeichnung dieses Programmabschnitts – unter anderem sichtbar in der Navigation. Für Produzenten ist in erster Linie der Download von Transferpaketen interessant, während die Administratoren viel mehr nach Paketen und ihren Statuswerten suchen. So erhalten Produzenten die Ausgabe der Bezeichnung „herunterladen“ und Administratoren die Bezeichnung „suche“, wobei die Programmaufgaben inhaltlich sehr ähnlich sind. Ein Produzent kann seinen eigenen Gesamtbestand anzeigen lassen und in detaillierten Suchergebnissen die Metadaten und den Bearbeitungsstatus einsehen. Die Suchergebnisse gliedern sich dabei in Projekt- und Ablieferungs-informationen. Wurde eine Datenlieferung erfolgreich vom Archivsystem zurückgeliefert, kann es als AIP vom Produzenten heruntergeladen werden. Die Administratoren erhalten an dieser Stelle andere Optionen. Sie können fehlerhafte Pakete löschen und für das Archivsystem vorbereitete Pakete in den Input verschieben.

Das Aufrufen der Suchergebnisse kann optional über die Anzeige des Gesamtbestandes oder durch die einfachen oder erweiterten Suchfunktionen stattfinden. Bei der Nutzung von Suchfunktionen wird die Ausgabe der Suchergebnisse in Archivpaketen (Projekten) und Datenlieferungen unterteilt. Die einfache Suche durchsucht dabei alle Datenfelder, die Informationen zu Projekten oder Ablieferungen enthalten. Die Eingabe des Suchwortes erfolgt über den Suchschlitz als Freitext. Die erweiterte Suchmaske bietet hingegen die Möglichkeit den Datenbestand auf spezifische Suchkriterien zu vergleichen. Für Produzenten und Administratoren sind die Auswahlmöglichkeiten identisch, mit Ausnahme der Institutionsauswahl, die nur Administratoren gestattet ist. Die Datenbank kann ansonsten auf folgende Datenfelder untersucht werden:

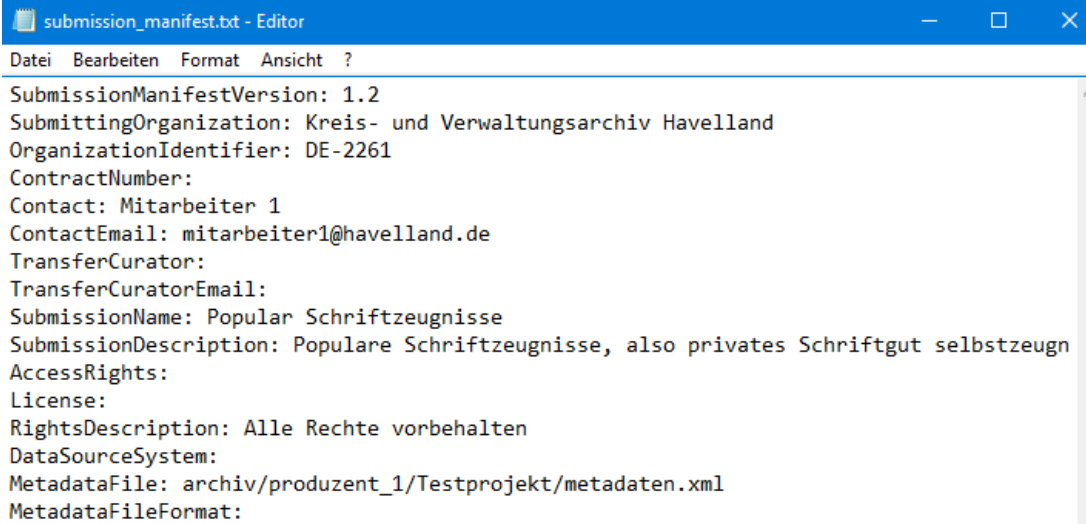
Suchoptionen	Beschreibung	Anmerkung
Suche beschränken auf	Beschränkung der Suche auf Projekte, Archivpaketen oder beidem	Radio-Buttons
Paket	Suche nach Paketbezeichnung	Freitext
Status	Suche nach Bearbeitungsstatus einer Ablieferung	Auswahlmenü
Projekt	Suche nach Projekttitle	Freitext
Projekt- und Datenlieferungsbeschreibung	Suche in den Beschreibungen eines Projektes und/oder Ablieferung	Freitext, Volltextsuche
Jahr	Suche nach dem Jahr der Projekterstellung und Ablieferungsdatum	Format yyyy
Art der Finanzierung	Suche nach einer bestimmten Finanzierung	Auswahlmenü
Server-ID	Suche nach ID-Nummer des Projektes und/oder Ablieferung	Ganze Zahl
Sortierung	Diverse Sortierungsmöglichkeiten der Suchergebnisse	Auswahlmenü

Tabelle 2: Datenfelder der erweiterten Suche

### 5. Programmabschnitt: Hochladen

Der letzte Programmabschnitt dient der Ablieferung von Transferpaketen. Skripte dieses Abschnittes werden im Verzeichnis „hochladen“ abgelegt. Produzenten können hier Projekte mit den dazugehörigen Metadaten erstellen. Beim Abgeben einer Datenlieferung kann das entsprechende Projekt ausgewählt und zugeordnet werden. Besitzt das abzuliefernde Transferpaket eine geringere Dateigröße im Vergleich zur serverzulässigen Maximalgröße, kann das Transferpaket direkt über den Aggregator hochgeladen werden (siehe Anhang G). Das Programm prüft dabei die Validität der Datenlieferung und ermittelt die technischen Metadaten. Der Produzent muss zudem deskriptive Metadaten in Form einer Beschreibung eingeben. Überschreitet das abzuliefernde Transferpaket jedoch die maximal zulässige Dateigröße des Servers, muss sich der Produzent mit der Administration in Verbindung setzen. Dafür wird vom Aggregator eine E-Mail-

Funktion bereitgestellt, wodurch automatisch relevante Metadaten an die Administration geliefert werden. Das führt zu einer Vereinfachung bei der Verwaltung und Zuordnung. Mangels eines Webservers im Testsystem wird auch diese E-Mail lediglich simuliert. Nach dem Kontakt zwischen Produzent und Administration müssen die Transferpakete manuell, zum Beispiel durch einen mobilen Datenträger, in das System gelangen (siehe Anhang H). Die Administration lädt die jeweiligen Daten diesbezüglich in das Verzeichnis „datentraeger“. Der Programmabschnitt „hochladen“ hat ebenfalls die Aufgabe das Verzeichnis „datentraeger“ auszulesen. Dies geschieht durch Aufruf des entsprechenden Skriptes, das den Produzenten verwehrt bleibt. Befinden sich Daten in diesem Verzeichnis, kann die Administration ein Transferpaket einem Projekt und somit einer Einrichtung zuordnen. Auch in diesem Prozess werden technische und deskriptive Metadaten erhoben. Eine weitere Aufgabe des Programmabschnittes ist die Erzeugung der Metadaten-Dateien. Während der Projekterstellung wird die „submission\_manifest.txt“ generiert. Die Datenstruktur richtet sich dabei nach der Mustervorlage für Übernahme-vereinbarungen<sup>9</sup> vom Zuse Institute Berlin (ZIB).



```

SubmissionManifestVersion: 1.2
SubmittingOrganization: Kreis- und Verwaltungsarchiv Havelland
OrganizationIdentifier: DE-2261
ContractNumber:
Contact: Mitarbeiter 1
ContactEmail: mitarbeiter1@havelland.de
TransferCurator:
TransferCuratorEmail:
SubmissionName: Popular Schriftzeugnisse
SubmissionDescription: Populare Schriftzeugnisse, also privates Schriftgut selbstzeugn
AccessRights:
License:
RightsDescription: Alle Rechte vorbehalten
DataSourceSystem:
MetadataFile: archiv/produzent_1/Testprojekt/metadaten.xml
MetadataFileFormat:

```

Abbildung 7: Beispielhafte Darstellung der Datei submission\_manifest.txt

Die zweite Metadaten-Datei ist die „metadaten.xml“. Diese wird beispielhaft in einem einfachen XML-Format valide und wohlgeformt generiert. In ihr werden alle Daten eines Projektes gespeichert und sie wird nach jedem neuen Hochladeprozess aktualisiert.

<sup>9</sup> siehe Klindt et al., 2016, S. 2-3

```

<?xml version="1.0" encoding="UTF-8"?>
- <archivpaket>
  <projekttitel>Liederbuch von Friesack</projekttitel>
  <projektbeschreibung>Populare Schriftzeugnisse, also privates Schriftgut selbstzeugnisähnlichen
    Charakters...</projektbeschreibung>
  <jahr>2017</jahr>
  - <institution>
    <institutionname>Kreis- und Verwaltungsarchiv Havelland</institutionname>
    <isil>DE-2261</isil>
  </institution>
  - <finanzierung>
    <finanzierungsart>Internes Projekt</finanzierungsart>
  </finanzierung>
  - <datenlieferungen>
    - <datenlieferung>
      <dateiname intern="transfer_1">liederbuch_von_friesack_1.zip</dateiname>
      <dateityp>zip</dateityp>
      <ablieferungsbeschreibung>Die insgesamt 27, vor allem Liebeslieder wurden überwiegend
        in Schönschrift aufgeschrieben. Es sind Liedtexte mit Themen wie: Die Liebe, Die
        Thräne, Die holde Nacht; aber auch: Rnaldini. (Teil 1 von 2)</ablieferungsbeschreibung>
      <dateigroesse einheit="MB">456</dateigroesse>
      <dateimenge>24</dateimenge>
    </datenlieferung>
    - <datenlieferung>
      <dateiname intern="transfer_2">liederbuch_von_friesack_2.zip</dateiname>
      <dateityp>zip</dateityp>
      <ablieferungsbeschreibung>Die insgesamt 27, vor allem Liebeslieder wurden überwiegend
        in Schönschrift aufgeschrieben. Es sind Liedtexte mit Themen wie: Die Liebe, Die
        Thräne, Die holde Nacht; aber auch: Rnaldini. (Teil 2 von 2)</ablieferungsbeschreibung>
      <dateigroesse einheit="MB">455</dateigroesse>
      <dateimenge>24</dateimenge>
    </datenlieferung>
  </datenlieferungen>
</archivpaket>

```

Abbildung 8: XML-Ausgabe der Datei metadaten.xml im Browser mit beispielhaftem Datensatz

Die Kommunikation zwischen der Webanwendung und dem Archivsystem wird in einer Simulation dargestellt. Diese befindet sich zu demonstrativen Zwecken in diesem Programmabschnitt. Die Ausführung kann nur durch Administratoren gestartet werden. Der Prozessablauf wird im nächsten Abschnitt erläutert.

## 4.5 Simulation des Archivierungsprozesses

Über das Suchergebnis eines Transferpakets kann die Administration die Übernahme in das Archivsystem veranlassen (siehe Anhang I). Dabei wird das jeweilige Transferpaket in das Input-Verzeichnis des Aggregators verschoben. Zudem wird das ZIP-komprimierte Transferpaket entpackt und die Metadaten-Dateien mitgeliefert. Das Skript „simulation.php“ simuliert und demonstriert den weiteren Verlauf der Übernahme in das digitale Archiv. Zunächst wird dazu der Input vom Aggregator ausgelesen und auf vorhandene Transferpakete untersucht. Parallel wird der Output-Ordner durchsucht. Mögliche vorhandene Dateien im Input und Output werden

separat gelistet. Sind alle notwendigen Dateien im Input vorhanden, kann per Knopfdruck die erste von zwei Phasen gestartet werden. Die erste Phase simuliert die Abholung aller Dateien vom Archivsystem, die Verarbeitung im Archivsystem in Form einer erneuten ZIP-Komprimierung der Einzeldateien und Zurücklieferung in das Output-Verzeichnis. Nach diesem Prozess befindet sich im Output eine einzelne ZIP-komprimierte Datei – der Input ist leer. In der zweiten Phase kann wieder auf Knopfdruck die Simulation fortgesetzt werden. Dabei wird das Transferpaket dem Produzenten und dem Projekt zugeordnet, gegebenenfalls umbenannt und als AIP zum Download zur Verfügung gestellt. Input und Output sind nun leer, womit der nächste Übernahmeprozess durchgeführt werden kann.

## **5 Schlussfolgerung und Ausblick**

Das entwickelte Aggregationsportal als Demonstrator für einen Pre-Ingest-Bereich ist prinzipiell funktionsfähig. Im entstandenen Zustand kann es als Lösungsansatz angesehen werden. Für eine tatsächliche Programmumsetzung sind weitere Faktoren zu berücksichtigen. In diesem Abschnitt soll retrospektiv auf das Ergebnis eingegangen und indessen die Probleme, Fehler, Verbesserungen und weitere Anforderungen angeführt werden.

### **5.1 Zielsetzung und Lösung**

In der Verbundlösung Berlin-Brandenburg werden digitale Kulturgüter von abliefernden Einrichtungen im Land Brandenburg in ein digitales Archiv transferiert. Unter Anwendung der Software Archivemtica nimmt die KBD Transferpakete in Form eines SIP entgegen. Dies geschieht in der Funktionseinheit Ingest, welcher im weiteren Prozess ein AIP erstellt. Das AIP kann wiederum an die digiS versandt und aufbewahrt werden. Das ausgearbeitete webbasierte Portal gliedert sich als Pre-Ingest in den Anfang dieses Verlaufes ein. Aus abliefernden Transferpaketen und Metadaten wird ein SIP für das digitale Archiv gebildet. Dieses enthält alle zu archivierenden Einzeldateien und Metadatendateien. Alle beinhalteten Elemente können vom Archivsystem kopiert werden. Die konzipierte Webanwendung kann diesen Prozess demonstrativ umsetzen und erfüllt demnach die Zielsetzung dieser Arbeit. Darüber hinaus treten weitere Fragen in Bezug des Programms und des gesamten Prozesses auf, die eine kritische Auseinandersetzung erfordern. Diesbezüglich sollen erst einmal Kriterien der Softwareentwicklung betrachtet werden.

## 5.2 Interaktion und Fehlerbehandlung

Da die Entwicklung der Webanwendung zunächst demonstrativen Zwecken dient, ignoriert diese partiell eine Vielzahl von Kriterien bezüglich Interaktionen zwischen den Nutzern und der Software. Beispiele hierfür sind informative Dialoge und angemessene Ausdruckweise, Konsistenz in der Gestaltung oder geeignete Rückmeldungen zum Geschehen (Dahm, 2006, S. 150-159). In diesem Abschnitt soll weniger auf den gesamten Themenbereich der Softwareergonomie eingegangen, sondern auf die Fehlerbehandlung hingewiesen werden. Viele mögliche Fehlerquellen oder Fehlersituationen werden von der prototypischen Portallösung nicht einwandfrei behandelt. Dies betrifft Programmfehler in der Software, Funktionsstörungen in der Hardware oder falsche Eingaben durch den Nutzer. Programmreaktionen auf Softwarefehler sind kaum möglich. Die Berücksichtigung auf externe Instanzen (Datenträger, Netzwerke, etc.) sollten hingegen immer eingeplant werden. Dabei können Fehler gemeldet, eventuell korrigiert oder sogar auf organisatorische, logische oder inhaltliche Aspekte hin vermieden werden. Eine Meldung als Reaktion auf einen Fehler sollte dem Nutzer entsprechend seines Benutzerstatus das Symptom, die Auswirkung, gegebenenfalls die Ursache und mögliche Korrekturhinweise liefern (Dahm, 2006, S. 237-239). Existierende Fehlermeldungen des Portals entsprechen nicht immer diesem Muster. Generell deckt die Anwendung bei weitem nicht alle Eventualitäten ab. Eine umfangreiche Analyse möglicher Fehler sollte in den Programmablauf integriert werden, um Fehler nach Möglichkeit von vornherein zu umgehen.

## 5.3 Hohe Datenmengen und Upload

Beim Hochladen von Transferpaketen wird das maximale Datenvolumen anhand der Systemvariablen in der Konfigurationsdatei „php.ini“ und dementsprechend durch den Webserver beschränkt. Ein Upload mit einer Dateigröße von mehr als 10 Gigabyte kann über ein Webinterface zu Problemen führen. Ein sicherer Upload hängt von einigen Faktoren ab: wie etwa eine stabile Internetverbindung, eine hohe Übertragungsrate oder die



Speicherkapazität. Prinzipiell ist es sinnvoll das erlaubte Maximum der Datenmenge einzuschränken. Es sollte vermieden werden, dass Hochladeprozesse zu viel Zeit benötigen oder sogar einen Timeout auslösen. Neben dem eigentlichen Upload finden Prüfzeiten und weitere interne Prozesse statt. Dies kann zum Beispiel eine Datenüberprüfung, eine Validierung oder die Datenverwaltung (verschieben, kopieren, umbenennen) sein. Um einen maximalen Grenzwert festzulegen, können einerseits HTML-Attribute definiert oder der Webserver PHP-seitig konfiguriert werden. Der Webaggregator versucht mittels der Datei „.htaccess“ vier Variablen anzupassen. Zunächst die Variablen „upload\_max\_filesize“ für die maximale Dateigröße und „post\_max\_size“ für den Grenzwert der zu übertragenden POST-Daten. Der zweite Wert sollte dabei immer größer sein als der Wert der maximalen Dateigröße, weil POST-Daten neben den Uploaddateien weitere übermittelte Daten enthalten. Des Weiteren kann durch die Variable „memory\_limit“ der Speicherverbrauch des Skriptes begrenzt und mittels „max\_execution\_time“ die Laufzeit des Skriptes angepasst werden. Für die Webanwendung besteht ein Vorteil in der Häufigkeit der Uploads. Es ist anzunehmen, dass diese gering sein wird. Eine Einschränkung bezüglich der maximalen Anzahl von Clients ist nicht zwingend notwendig. Ein Nachteil besteht wiederum darin, dass die Transferpakete mutmaßlich höhere Dateigrößen aufweisen, als dass sie durch das Portal übertragen werden können. Alternativen, wie etwa die Übertragung der Daten unter Einsatz eines mobilen Datenträgers oder einer FTP-Übertragung, sind notwendig. Zusätzlich fordern mehrfache Speicherungen von digitalen Kopien höhere Speicherkapazitäten, im Besonderen beim Übernahmeprozess in das Archivsystem. Zur Entlastung von Speicherressourcen können die Duplikate des AIPs zeitweilig abgespeichert und nach einer Frist gelöscht werden. Wird ein Transferpaket nach der Löschung benötigt, so kann dieses vom digitalen Archiv wiederhergestellt werden. Hierfür sind Funktionalitäten zum Anfordern eines bereits entfernten Transferpakets und ein entsprechender Bearbeitungsstatus notwendig.

## 5.4 Metadaten und modulare Erweiterungen

Eine Weiterentwicklung hinsichtlich der Erhebung relevanter Metadaten sollte für die Anwendung in Analogie zur Datenbank erfolgen. Eine noch striktere Prüfung und Validierung eingegebener Werte kann durch HTML, PHP und Java stattfinden, wobei auch reguläre Ausdrücke eingesetzt werden können. Die erfassten Metadaten werden simultan in der Datenbank, in der Submission Manifest und einer XML-Datei gespeichert. Letzteres wird von der Webanwendung provisorisch in einem einfachen Schema umgesetzt. In Abhängigkeit zum eingesetzten Archivsystem werden eventuell weitere Dateien (Protokolle, Dokumentierungen, Prüfsummen, etc.) gebraucht. Auch alternative Formate (CSV, etc.) können auf Eignung untersucht werden. In Bezug auf XML können mittels XML-Technologien Standardformate durch Mapping erzeugt werden: ein Beispiel hierfür ist das Schemata METS. Durch entsprechende Technologien und Werkzeuge ist die Erstellung eines spezifischen XML-Formats über eine Shell möglich. Daraus ergibt sich, dass am Beispiel einer METS-Datei in Frage kommende Editoren auf Kommandozeilenebene eine METS-Struktur erzeugen können. Ein Tool zur Erzeugung einer XML-Datei ist nur eine Einsatzmöglichkeit für eine Anwendungserweiterung durch Werkzeuge. Im Bereich der digitalen Langzeitarchivierung wurden bereits einige Softwareentwicklungen realisiert. Viele Programme stehen dabei quelloffen zur Verfügung und können den individuellen Anforderungen angepasst werden. Innerhalb des Aggregationsportals können durch externe Software zusätzliche Funktionen integriert werden. In diesem Zusammenhang ist es möglich solche Programme modular einzubetten und sie auf diese Weise vom Portal ausgehend serverbasierend anzusteuern. Die Erweiterungsmöglichkeiten können sich dabei über den gesamten Pre-Ingest-Bereich erstrecken. Manche Programme dienen vorrangig der Strukturierung innerhalb der hierarchischen Unterverzeichnisse der Webanwendung oder der generellen Paketstruktur. Andere Softwareanwendungen können die Erstellung und Ablieferung von Informationspaketen unterstützen oder auch den Übernahmeprozess selbst. Zusätzliche Dateien können nach den gegebenen Anforderungen in komparablen Codierungen generiert werden. Durch

etwaige Erweiterungen können demzufolge einheitliche und möglichst vollständige SIPs für das digitale Archiv bereitgestellt werden, womit sich Störfaktoren wie etwa Zeitaufwand oder Mehrarbeit für die Administration reduzieren. Dementsprechend optimieren modulare Funktionserweiterungen den gesamten Workflow durch externe Applikationen. Ferner können Module und Erweiterungen der Skriptsprache PHP aus vorhandenen Bibliotheken eingebunden werden, um die Anwendung zu verbessern. Dazu zählen unter anderem die Installationsmechanismen, die Paketverwaltung und die Gewährleistung der Informationssicherheit. Unabdingbar sollten Sicherheitsmaßnahmen in die Entwicklung eines Pre-Ingest-Werkzeuges berücksichtigt werden. Dies betrifft eingebundene computerbasierte Systeme und Backupmechanismen, sowie den Datenschutz und die Datensicherheit.

## Fazit

Im Rahmen dieser Bachelorarbeit wurden zunächst die Herausforderungen in der digitalen Langzeitarchivierung aufgezeigt. Im Besonderen wurde auf fehlende oder unzureichend beschriebene Standards hingewiesen. Standardisierungsprozesse sind für eine einheitliche Zusammenführung von Dateien und Metadaten in Digitalisierungsprojekten unabdingbar. Dieser Sachverhalt wird ebenfalls am Beispiel der bisherigen Projekte im Land Brandenburg deutlich. Die dortigen Archivierungsvorhaben richten sich nach dem OAIS-Funktionsmodell. Dieses beginnt mit der Funktionseinheit INGEST und weist fehlende Standardisierungen für die vorangegangenen Prozesse auf. Das SIP, welches als Übernahmepaket in den INGEST-Bereich eines Archivsystems gelangt, soll durch einen PRE-INGEST in vereinheitlichter Form erstellt werden. Für dieses Vorhaben wurde im Verlauf dieser Arbeit eine webbasierte Portallösung konzipiert und entwickelt. Das Portal basiert auf einer relationalen Datenbank. Diese wurde auf die Anforderungen einer prototypischen Lösung analysiert, modelliert und programmtechnisch in eine Webanwendung integriert. Die webbasierte Anwendung wurde mithilfe von Auszeichnungs-, Programmier- und Datenbanksprachen umgesetzt. Die am Ende entstandene Portallösung dient demonstrativen Zwecken und kann die grundlegende Funktionsweise eines Pre-Ingest-Werkzeuges darstellen. Die Auswertung ergab, dass in der Entwicklung vielzählige Faktoren zu berücksichtigen sind, um ein Programm für den realen Betrieb zu entwerfen. Des Weiteren wurden spezifische Herausforderungen genannt, wie zum Beispiel die hohen Datenmengen bei der digitalen Langzeitarchivierung, die Konzeption der Metadaten-dateien oder die modularen Erweiterungen, welche das System ergänzen oder verbessern. Die prototypische webbasierte Anwendung ist keine vollständige und stabile Software. Sie erfüllt ihren Zweck als Demonstrator und kann als Grundlage für weitere Entwicklungsprozesse verwendet werden.

## Abbildungsverzeichnis

<b>ABBILDUNG 1:</b> OAIS-FUNKTIONSMODELL NACH ISO 14721:2012.....	10
<b>ABBILDUNG 2:</b> AUSZUG DES EINGABEFORMULARS ZUM ISIL WÄHREND DER PRODUZENTENREGISTRIERUNG IM DEMONSTRATOR .....	13
<b>ABBILDUNG 3:</b> SYSTEMARCHITEKTUR DES DEMONSTRATORS .....	15
<b>ABBILDUNG 4:</b> VERZEICHNISSTRUKTUR ZUR SPEICHERUNG VON TRANSFERPAKETEN IM DEMONSTRATOR .....	27
<b>ABBILDUNG 5:</b> WIREFRAME DES DEMONSTRATORS .....	29
<b>ABBILDUNG 6:</b> SCREENSHOT DER STARTSEITE DES DEMONSTRATORS NACH DER ANMELDUNG DES SYSTEMADMINISTRATORS.....	32
<b>ABBILDUNG 7:</b> BEISPIELHAFTE DARSTELLUNG DER DATEI SUBMISSION_MANIFEST.TXT .....	36
<b>ABBILDUNG 8:</b> XML-AUSGABE DER DATEI METADATEN.XML IM BROWSER MIT BEISPIELHAFTEM DATENSATZ .....	37

## Tabellenverzeichnis

<b>TABELLE 1:</b> ZUGRIFFSRECHTE AUF PROGRAMMABSCHNITTE NACH NUTZERGRUPPEN ..	32
<b>TABELLE 2:</b> DATENFELDER DER ERWEITERTEN SUCHE .....	35

## Literaturverzeichnis

Brübach, N. (2010). Das Referenzmodell OAIS. In H. Neuroth, A. Oßwald, R. Scheffel, S. Strathmann & K. Huth (Hrsg.), *Eine kleine Enzyklopädie der digitalen Langzeitarchivierung* (Version 2.3, Kap. 4:9-4:10). Zugriff am: 17.02.2017, verfügbar unter: <http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:0008-2010071949>

Dahm, M. (2006). *Grundlagen der Mensch-Computer-Interaktion*. München: Pearson Studium.

DIN 31644 (2012). *Information und Dokumentation. Kriterien für vertrauenswürdige digitale Langzeitarchive*. Berlin: Beuth Verlag GmbH.

Dobratz, S. & Tappenbeck, I. (2002). Thesen zur Zukunft der digitalen Langzeitarchivierung in Deutschland. In A. Kellersohn et al., *Bibliothek. Forschung und Praxis* (Band 26, S. 257-261). Berlin: De Gruyter.

Faeskorn-Woyke, H., Bertelsmeier, B., Riemer, P. & Bauer, E. (2007). *Datenbanksysteme. Theorie und Praxis mit SQL2003, Oracle und MySQL*. München: Pearson Deutschland GmbH.

ISO 15511 (2011). *Information and documentation. International standard identifier for libraries and related organizations (ISIL)*. Berlin: Beuth Verlag GmbH.

Keitel, C. & Schoger, A. (2013). *Vertrauenswürdige digitale Langzeitarchivierung nach DIN 31644* (1. Aufl.). Berlin: Beuth Verlag GmbH.

- Klimpel, P. (2013). Was bleibt?. In J. Klimpel & J. Keiper (Hrsg.), *Was Bleibt? Nachhaltigkeit der Kultur in der digitalen Welt* (1. Aufl., S. 11-14). Berlin: iRights.Media.
- Klindt, M., Amrhein, K., Hasler, T., Oltmanns, E. & Peters-Kottig, W. (2016). *Digitale Langzeitarchivierung: Mustervorlage für Übernahmevereinbarungen*. Berlin: Zuse Institute Berlin. Zugriff am: 15.02.2017, verfügbar unter: <http://nbn-resolving.de/urn:nbn:de:0297-zib-59452>
- nestor-Arbeitsgruppe OAIS-Übersetzung/Terminologie (2013). *Referenzmodell für ein Offenes Archiv-Informationssystem*. Zugriff am: 15.02.2017, verfügbar unter: <http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:0008-2013082706>
- Qin, J. & Zeng, M. L. (2016). *Metadata* (2. Aufl.). Chicago: American Library Association.
- Rehm, G. (2013). Eine Strategie zur Förderung der digitalen Langzeitarchivierung. In J. Klimpel & J. Keiper (Hrsg.), *Was Bleibt? Nachhaltigkeit der Kultur in der digitalen Welt* (1. Aufl., S. 199-214). Berlin: iRights.Media.
- Steffenhagen, B. (2016). *Die Erweiterung des OAIS-Referenzmodells um die Funktionseinheit Pre-Ingest am Beispiel der Verbundlösung Berlin-Brandenburg. Konzeption einer Web-Anwendung zur Vorbereitung der Übernahme von Informationspaketen*. Unveröffentlichte Masterarbeit, Fachhochschule Potsdam.

## Anhang

**ANHANG A:** Identifizierung von Objekttypen und Objekteigenschaften

**ANHANG B:** Bestimmung der paarweisen Objektbeziehungen

**ANHANG C:** ER-Modell für die Datenbank des Demonstrators

**ANHANG D:** Relationales Datenbankmodell (RDM) des Demonstrators

**ANHANG E:** Verzeichnisstruktur der Webanwendung

**ANHANG F:** Nutzerübersichten und Registrierung

**ANHANG G:** Transferpakete hochladen und ausgeben

**ANHANG H:** Datenlieferungen manuell hinzufügen

**ANHANG I:** Simulation des Archivierungsprozesses

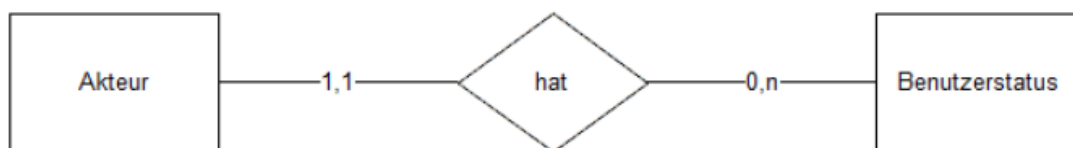


## ANHANG A: Identifizierung von Objekttypen und Objekteigenschaften

Name	Art	Beschreibung
Akteur	Objekttyp	Tabelle mit den Akteuren im System
#akteur_id	Objekteigenschaft	In Tabelle Akteur der zugeteilte und einzigartige Primärschlüssel
benutzername	Objekteigenschaft	Benutzername des Akteurs, optional für Administratoren
passwort	Objekteigenschaft	Passwort des Akteurs zur Anmeldung in das System
email	Objekteigenschaft	E-Mail-Adresse des Akteurs
telefonnummer	Objekteigenschaft	Telefonnummer des Akteurs
Benutzerstatus	Objekttyp	Tabelle aller vorhandenen Benutzerstatus zur Zuordnung
#benutzerstatus_id	Objekteigenschaft	In Tabelle Benutzerstatus der zugeteilte und einzigartige Primärschlüssel
bezeichnung	Objekteigenschaft	Benennung des Benutzerstatus
Institution	Objekttyp	Tabelle der Institutionsdaten
#ISIL	Objekteigenschaft	Sigel der Institution als zugeteilter und einzigartiger Primärschlüssel
ISIL_link	Objekteigenschaft	Link zum Datensatz der Staatsbibliothek zu Berlin über ISIL
institutionsname	Objekteigenschaft	Name der Einrichtung
ansprechpartner	Objekteigenschaft	Administrativer Ansprechpartner der Einrichtung
ansprechpartner_email	Objekteigenschaft	E-Mail-Adresse des administrativen Ansprechpartners der Einrichtung
Archivpaket	Objekttyp	Tabelle der erstellten Projekte einer Institution
#archivpaket_id	Objekteigenschaft	In Tabelle Archivpaket der zugeteilte und einzigartige Primärschlüssel
projekttitel	Objekteigenschaft	Bezeichnung des Projektes
projektbeschreibung	Objekteigenschaft	Beschreibung des Inhalts des jeweiligen Projekts
jahr	Objekteigenschaft	Jahr der Erstellung
kurator	Objekteigenschaft	Name des Projektverantwortlichen, der die Daten bereitstellt
kurator_email	Objekteigenschaft	E-Mail-Adresse des Projektverantwortlichen
rechte	Objekteigenschaft	Rechtliche Angaben zum Projekt
paketverzeichnis	Objekteigenschaft	Zielverzeichnis für die Daten des Projektes im System
Datenlieferung	Objekttyp	Tabelle zur Speicherung der Metadaten für abgelieferte Dateien
#datenlieferung_id	Objekteigenschaft	In Tabelle Datenlieferung der zugeteilte und einzigartige Primärschlüssel
dateiname	Objekteigenschaft	Ursprünglicher Dateiname der Ablieferung
dateiname_intern	Objekteigenschaft	Zur systeminternen Verwaltung der Datei vergebener Dateiname
dateityp	Objekteigenschaft	Typ der Datei (z. B. ZIP)
ablieferungsbeschreibung	Objekteigenschaft	Kurze Beschreibung der Ablieferung
ablieferungsdatum	Objekteigenschaft	Hochladedatum der Ablieferung
ablieferungszeit	Objekteigenschaft	Hochladeuhrzeit der Ablieferung
dateigroesse	Objekteigenschaft	Anzahl der Bytes einer Ablieferungsdatei
dateimenge	Objekteigenschaft	Menge der Einzeldateien im verpackten Abgabeformat
Bearbeitungsstatus	Objekttyp	Tabelle zur Zuordnung des aktuellen Standes einer Ablieferung
#bearbeitungsstatus_id	Objekteigenschaft	In Tabelle Bearbeitungsstatus der zugeteilte und einzigartige Primärschlüssel
bezeichnung	Objekteigenschaft	Bezeichnung des Status
beschreibung	Objekteigenschaft	Beschreibung des Status
Finanzierungsart	Objekttyp	Tabelle zur Speicherung vorhandener Finanzierungsarten für Projekte
#finanzierungsart_id	Objekteigenschaft	In Tabelle Finanzierungsart der zugeteilte und einzigartige Primärschlüssel
art	Objekteigenschaft	Bezeichnung der Finanzierungsart

## ANHANG B: Bestimmung der paarweisen Objektbeziehungen

### 1. Relation: Akteur – Benutzerstatus



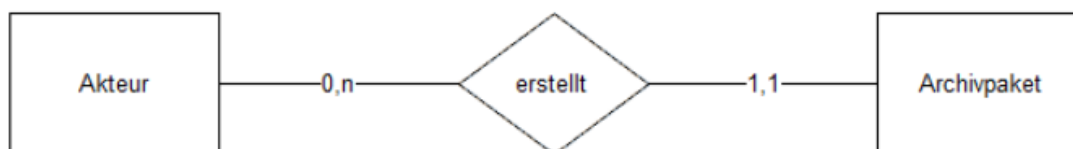
Einem Akteur muss genau ein Status zugewiesen werden. Ein bestimmter Status (Produzent oder Administrator) kann beliebig vielen Akteuren zugeordnet werden (unverzüglich nach der Installation auch keinem).

### 2. Relation: Akteur – Institution



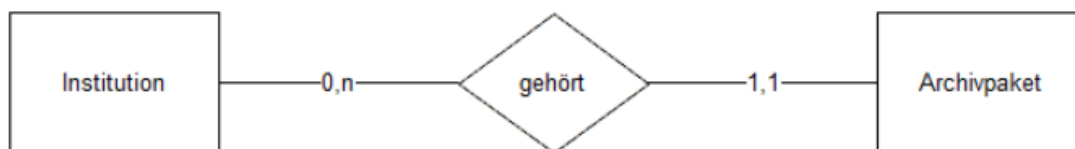
Einem Akteur kann maximal eine Institution zugewiesen werden, während eine Institution genau einem Akteur zugewiesen werden muss.

### 3. Relation: Akteur – Archivpaket



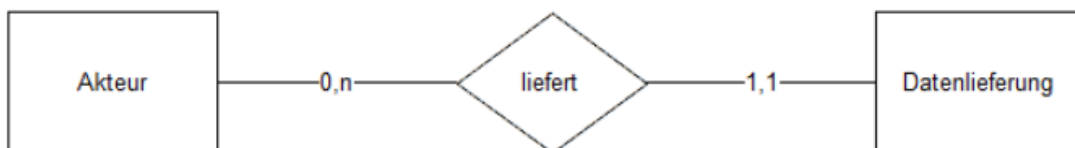
Ein Akteur kann beliebig viele Archivpakete beziehungsweise Projekte im System anlegen. Ein Archivpaket muss dabei genau dem Akteur, welcher das Projekt erstellt, zugewiesen werden.

## 4. Relation: Institution – Archivpaket



Identisch mit der Relation „Akteur – Archivpaket“ können einer Institution beliebig viele Archivpakete angehören. Ein Archivpaket muss dabei genau der Institution zugeordnet werden, für die das jeweilige Projekt erstellt wurde.

## 5. Relation: Akteur – Datenlieferung



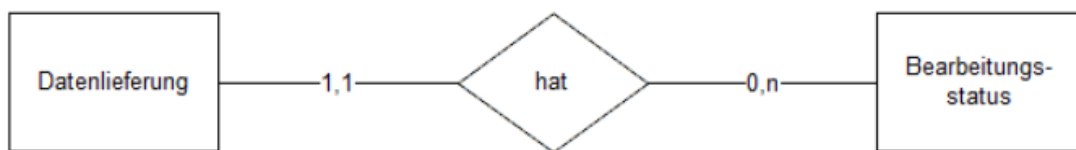
Ein Akteur kann beliebig viele Datenlieferungen im entsprechenden Format abliefern. Eine Datenlieferung muss dabei genau dem Akteur, der die Ablieferung ausgeführt hat, zugeordnet werden.

## 6. Relation: Datenlieferung – Archivpaket



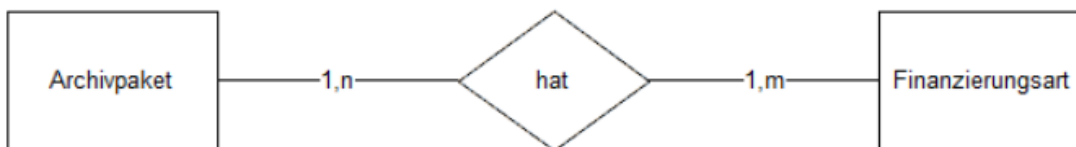
Eine Datenlieferung gehört genau einem Archivpaket an. Ein Archivpaket beziehungsweise Projekt kann beliebig viele Datenlieferungen besitzen.

## 7. Relation: Datenlieferung – Bearbeitungsstatus



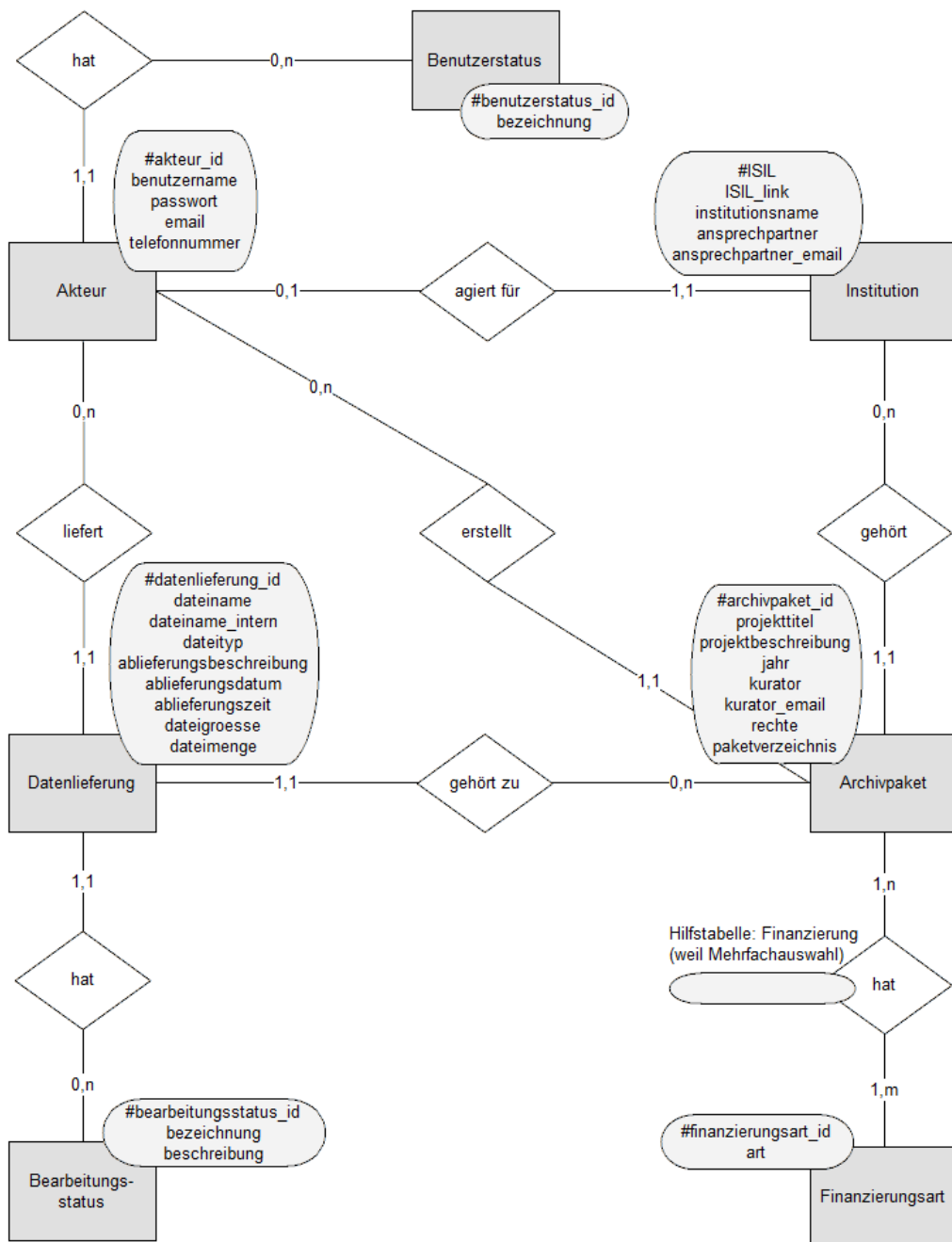
Eine Datenlieferung hat immer einen vordefinierten Bearbeitungsstatus, während ein bestimmter Status (z. B. AIP oder Archivierungsfehler) beliebig vielen Ablieferungen zugewiesen werden kann. Nicht jeder Status muss zur gleichen Zeit vergeben werden.

## 8. Relation: Archivpaket – Finanzierungsart

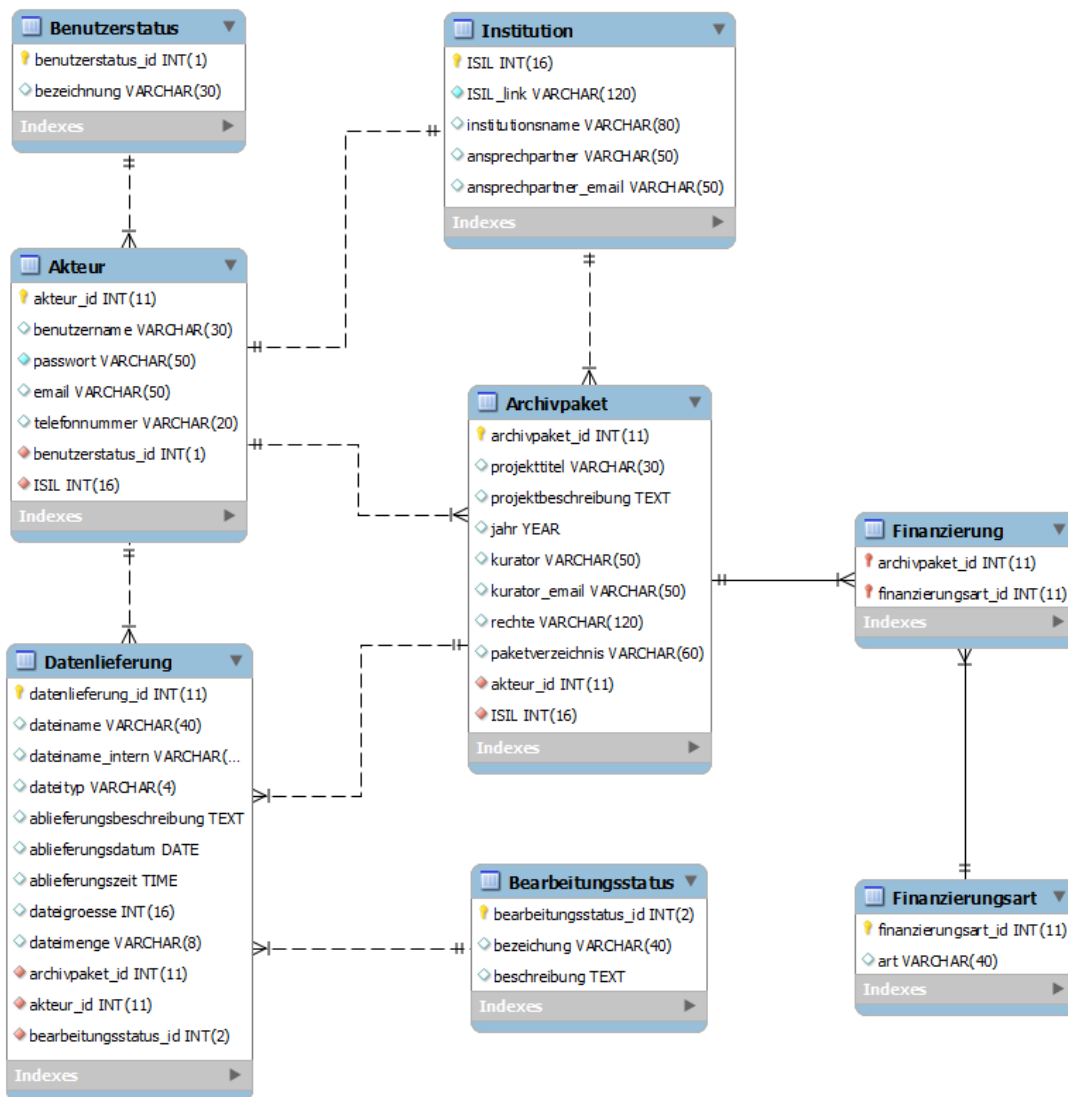


Ein Archivpaket hat mindestens eine, aber beliebig viele Angaben zu deren Finanzierung (z. B. ein internes Projekt oder eine Landesförderung). Eine Art der Finanzierung kann beliebig vielen Archivpaketen zugeordnet werden.

**ANHANG C: ER-Modell für die Datenbank des Demonstrators**

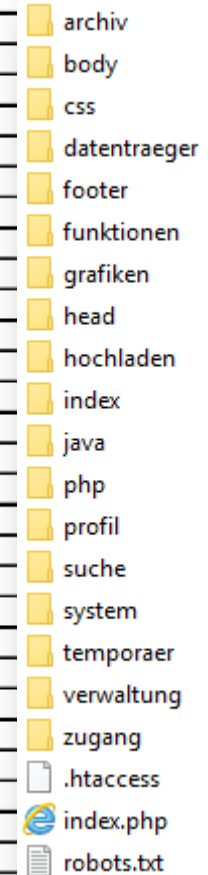


## ANHANG D: Relationales Datenbankmodell (RDM) des Demonstrators



## ANHANG E: Verzeichnisstruktur der Webanwendung

Nr.	Dateien				Kurzbeschreibung
	1.	2.	3.	4. Hierarchieebene	
1	archiv				Verzeichnis: Enthält die Datenlieferung und Metadaten aller Produzenten
2		produzent [*]			Verzeichnis eines Produzenten (pro Produzent ein Verzeichnis)
3			[*PROJEKT*]		Verzeichnis eines Projektes (pro Produzent beliebig viele Projekte)
4			metadaten.xml		Metadatendatei im XML-Format
5			submission_manifest.txt		Metadatendatei nach Übereinkommenvereinbarung
6			transfer [*].zip		Datenlieferung (beliebig viele pro Projekt)
7	body				Verzeichnis: Enthält Grundelemente des HTML-Body-Elements
8		footer.php			Fußzeile der Webseite
9		header.php			Bild und dazugehörige Verlinkung im Kopfbereich der Webseite
10		navigation.php			Navigation der Webseite
11		statuszeile.php			Obere Statusleiste auf der Webseite
12	css				Verzeichnis: Enthält CSS-Dateien für das Webseitendesign
13		dropdown.css			Gestaltung der Dropdown-Menü-Boxen
14		drucken.css			Gestaltung der Druckansicht (z. B. für Suchergebnisse)
15		elemente.css			Gestaltung von HTML-Elementen
16		klassen.css			Gestaltung von selbstdefinierten Klassen
17		main.css			Gestaltung aller Hauptboxen des Grunddesigns und Import aller weiteren CSS-Dateien
18		schalter.css			Gestaltung aller Navigationsschalter
19	datentraeger				Verzeichnis: Zielverzeichnis für manuell hochzuladene Datenlieferungen
20		datentraeger_hinweis.txt			Enthält Hinweise zu diesem Verzeichnis
21			[*TRANSFER*].zip		Ggf. manuell hochgeladenes Transferpaket (beliebig viele)
22	footer				Verzeichnis: Enthält Webseiten, welche im Footer verlinkt sind
23		faq.php			Webseite zum FAQ (Häufig gestellte Fragen)
24		faq.txt			Inhalt für die Webseite zum FAQ (Häufig gestellte Fragen)
25		footer_hinweis.txt			Enthält Hinweis auf Lokalisation des Skripts der eigentlichen Footer-Konfiguration
26		impressum.php			Webseite zum Impressum
27		impressum.txt			Inhalt für das Impressum
28		kontakt.php			Webseite zum Kontakt
29		kontakt.txt			Inhalt für Kontakt
30	funktionen				Verzeichnis: Enthält PHP-Funktionen
31		bytes_umrechnen.php			PHP-Funktion: Rechnet Bytes ggf. in höhere Einheiten um (GB, TB, etc.)
32		funktionsbibliothek.php			Sammelskript aller PHP-Funktionen (wird in jede Webseite eingebunden)
33		metadaten_xml.php			PHP-Funktion: Erzeugt eine XML-Datei mit Metadaten für ein Projekt
34		mysql_datenbank.php			PHP-Funktion: Funktionen für Abfragen und Ausführungen in der MySQL-Datenbank
35		passwort_generator.php			PHP-Funktion: Generiert ein zufälliges Passwort
36		rekursiv_loeschen.php			PHP-Funktion: Löscht ein Verzeichnis rekursiv
37		submission_manifest.php			PHP-Funktion: Erzeugt eine submission_manifest.txt Datei mit Metadaten



Nr.	Dateien				Kurzbeschreibung
	1.	2.	3.	4. Hierarchieebene	
38		text	umwandeln.php		PHP-Funktion: Funktionen zur Umwandlung bestimmter Textstellen und/oder Zeichen
39	<b>grafiken</b>				Verzeichnis: Bilddateien der Webseite
40		banner	.jpg		Bild (Banner) im Kopfbereich (Header) der Webseite
41		favicon	.gif		Favicon der Webseite
42	<b>head</b>				Verzeichnis: Enthält alle Einbindungen im HTML-Head-Element
43		dublin	core.php		Metadaten der Webseite nach Dublin Core
44		favicon	.php		Einbindung des Favicon
45		html	config.php		Diverse Konfigurationen (Zeichenkodierung, Webcrawler, etc.)
46		html	imports.php		Einbindungen von CSS- und Java-Dateien
47	<b>hochladen</b>				Verzeichnis: Enthält Webseiten des Hauptmenüpunktes "Hochladen"
48		hochladen	.php		Menüauswahl der "Hochladen"-Webseiten
49		paket	hochladen.php		Skript zum Hochladen einer Datenlieferung
50		paket	zuordnen.php		Skript zur Zuordnung manuell hochgeladener Datenlieferungen
51		projekt	erstellen.php		Skript zur Erstellung eines Projektes (Archivpaketes)
52		simulation	.php		Demonstratives Skript zur Simulation des Archivsystem-Vorgangs
53	<b>index</b>				Verzeichnis: Enthält Inhalte und relevante Skripte für die index.php-Datei
54		aktuelles	.txt		Inhalt der Informationsbox auf der Startseite
55		aktuelles	aktiv.txt		Status der Informationsbox auf der Startseite (aktiviert/deaktiviert)
56		begrueessungstext	.txt		Inhalt des Begrüßungstextes auf der Startseite
57		weiterleitung	.php		Skript zur Weiterleitung auf die Startseite, wenn erforderlich
58	<b>java</b>				Verzeichnis: Enthält Javaskripte
59		dropdown	menue.js		Java-Skript: Funktion des Dropdown-Menüs
60	<b>php</b>				Verzeichnis: Enthält weitere PHP-Skripte
61		anmeldestatus	.php		PHP-Skript: Legt Variablen des Akteurs fest
62		mysql	.php		PHP-Skript: Legt Variablen des Datenbankzugriffs fest
63		skriptbibliothek	.php		Sammelskript aller PHP-Skripte (wird in jede Webseite eingebunden)
64		zeit	.php		PHP-Skript: Legt aktuelle, zeitliche Variablen fest (Datum, Uhrzeit, etc.)
65	<b>profil</b>				Verzeichnis: Enthält Webseiten des Hauptmenüpunktes "Profil"
66		passwort	aendern.php		Skript zum Ändern des Passwortes eines Akteurs
67		profil	.php		Menüauswahl der "Profil"-Webseiten
68		profil	daten.php		Skript zur Anzeige der Profildaten
69	<b>suche</b>				Verzeichnis: Enthält Webseiten des Hauptmenüpunktes "Herunterladen" bzw. "Suche"
70		gesamtbestand	.php		Skript zur Anzeige aller Projekt und Datenlieferungen eines Produzenten
71		metadaten	aktualisieren.php		Skript zur Aktualisierung der metadaten.xml-Datei
72		paketsuche	.php		Skript zur Suche nach Datenlieferungen
73		paketsuche	erweitert.php		Skript zum Formular der erweiterten Suche nach Datenlieferungen
74		suche	.php		Menüauswahl der "Herunterladen"- bzw. "Suche"-Webseiten



Nr.	Dateien				Kurzbeschreibung
	1.	2.	3.	4. Hierarchieebene	
75		suchergebnis.php			Skript zur Anzeige aller Daten eines Projektes oder einer Datenlieferung
76	<b>system</b>				Verzeichnis: Enthält Webseiten des Hauptmenüpunktes "System"
77		datenbank_installieren.php			Skript zur Installation der Datenbank
78		datenbank_loeschen.php			Skript zur Deinstallation der Datenbank
80		fehler.pdf			PDF: Enthält eine Auflistung systemeigener Fehlermeldungen (Ursache und Lösung) -> unvollständig / fehlt
81		php_info.php			Skript zur Anzeige der PHP-Informationen des Servers
82		rdm.pdf			PDF: Enthält das relationale Datenmodell für den Aufruf über den Browser
83		system.php			Menüauswahl der "System"-Webseiten
84		verzeichnisstruktur.pdf			PDF: Enthält die Verzeichnisstruktur für den Aufruf über den Browser
85	<b>temporaer</b>				Verzeichnis: Enthält ggf. Dateien, welche mit dem Archivsystem interagieren (speichert temporär Dateien)
86	<b>input</b>				Verzeichnis: Enthält ggf. Daten für das Archivsystem
87		metadaten.xml			Gegebenenfalls: Metadaten-XML-Datei eines Projektes
88		submission_manifest.txt			Gegebenenfalls: submission_manifest.txt-Datei eines Projektes
89		[*DATEI*]			Gegebenenfalls: Entpackte Daten einer Datenlieferung (beliebig viele)
90	<b>output</b>				Verzeichnis: Enthält ggf. Daten vom Archivsystem
91		transfer [*]_aip.zip			Gegebenenfalls: Zurücklieferung einer neu verpackten und vom Archivsystem verarbeiteten Datenlieferung
92		transfersperre.txt			Enthält den Wert zur Zulassung oder Sperrung eines neuen Übernahmeprozesses (aktiviert/deaktiviert)
93	<b>verwaltung</b>				Verzeichnis: Enthält Webseiten des Hauptmenüpunktes "Verwaltung"
94		administratoren_uebersicht.php			Skript zur Übersicht aller Konten der Administratoren
95		administratorenkonto_einrichten.php			Skript zum Einrichten eines neuen Administratorenkontos
96		faq_bearbeiten.php			Skript zur Bearbeitung der FAQ Webseite (Footer)
97		finanzierungsarten.php			Skript zur Bearbeitung vorhandener Finanzierungsarten
98		impressum_bearbeiten.php			Skript zur Bearbeitung der Impressum Webseite (Footer)
99		kontakt_bearbeiten.php			Skript zur Bearbeitung der Kontakt Webseite (Footer)
100		produzenten_registrierung.php			Skript zum Einrichten eines neuen Produzentenkontos
101		produzenten_uebersicht.php			Skript zur Übersicht aller Konten der Produzenten
102		startseite_bearbeiten.php			Skript zur Bearbeitung der Startseite (Begrüßungstext und optionale Informationsbox)
103		verwaltung.php			Menüauswahl der "Verwaltung"-Webseiten
104	<b>zugang</b>				Verzeichnis: Enthält Skripte zum An- und Abmeldeprozess
105		login.php			Skript zur Anmeldung in das System (Login)
106		logout.php			Skript zur Abmeldung aus dem System (Logout)
107		passwort_admin.txt			Enthält Passwort des Systemadministratorenkontos (Wichtig zur Nutzung des Kontos)
108		passwort_vergessen.php			Skript zum E-Mail-Versand eines neuen Passwortes, wenn der Nutzer das Passwort vergessen hat
109		passwort_vergessen_aktivierung.php			Skript zur Aktivierung des neuen Passwortes
110		.htaccess			Konfigurationen der php.ini-Datei, sofern der Server dies zulässt
111		index.php			Skript der Startseite des gesamten Portals
112		robots.txt			Anweisungen bzw. Beschränkungen zur Verzeichnis- und Skriptdurchsuchungen für Webcrawler etc.

## ANHANG F: Nutzerübersichten und Registrierung

Die Registrierung gliedert sich in:

- Eingabeformular
- Eingabeüberprüfung (siehe Abbildung)
- Ausführung

Willkommen! Sie sind angemeldet als Administrator Abmelden



Startseite → Verwaltung → Neuen Produzenten registrieren

### Neuen Produzenten registrieren

Hochladen Suche Profil Verwaltung

Überprüfen Sie bitte Ihre Angaben. Kontrollieren Sie auch ob der erzeugte ISIL-Link zur "Staatsbibliothek zu Berlin" den entsprechenden Datensatz wiedergibt. Klicken Sie auf "Zurück" um Änderungen an den Daten durchzuführen. Sind alle Eingaben korrekt, klicken Sie anschließend auf "Registrieren" um ein neues Konto zu erstellen.

Kontodaten

**Ansprechpartner:** Frau Test  
**Institutionsname:** Archiv ABC  
**ISIL:** DE-ABC-1234  
**ISIL-Link:** <http://sigel.staatsbibliothek-berlin.de/nc/suche/?isil=DE-ABC-1234>  
**E-Mail-Adresse:** test@archiv.de  
**Telefonnummer:** 01 62123456  
**Passwort:** \* sl5qndhw

\* Bitte notieren Sie sich das Passwort vor der Registrierung: Das Passwort wird verschlüsselt und kann nicht wieder entschlüsselt werden

Registrieren Zurück

Copyright © 2017 FH Potsdam. Alle Rechte vorbehalten. FAQ • Kontakt • Impressum

Schritt 2 der Registrierung: Überprüfung der Formulareingaben

In der Produzentenübersicht werden alle registrierten Produzenten aufgelistet. Zudem bietet dieser Bereich eine Suche nach Produzenten.

Startseite → Verwaltung → Produzenten Übersicht

### Produzenten Übersicht

Hochladen Suche Profil Verwaltung

Suche

Name der Institution  ISIL  E-Mail-Adresse  Ansprechpartner  Suchen

Produzenten: 2

Institution	ISIL	Ansprechpartner	E-Mail-Adresse	Entfernen
Archiv ABC	DE-ABC-1234	Frau Test	test@archiv.de	löschen
Museum XYZ	DE-XYZ-1234	Herr Test	test@museum.de	löschen

Kontenübersicht aller Produzenten

Übersicht aller Produzentenkonten

Das Verfahren ist für Administratorenkonten ähnlich.

## ANHANG G: Transferpakete hochladen und ausgeben

Für dieses Beispiel wurde das Projekt „Testprojekt 01“ erstellt. Dem Projekt soll eine Datenlieferung durch den Produzenten hinzugefügt werden.

### Schritt 1: Projekt- und Optionsauswahl

Willkommen! Sie sind angemeldet als Museum XYZ Abmelden



Startseite → Hochladen → Archivpaket hochladen

## Archivpaket hochladen

Hochladen Herunterladen Profil

Wählen Sie das Projekt aus, für welches Daten hinzugefügt werden sollen.

Projekt (Archivpaket):

Wie soll Ihr Paket abgeliefert werden?

**BITTE BEACHTEN SIE:** Möchten Sie Ihr Paket direkt hochladen, darf die maximale Dateigröße von **8GB** nicht überschritten werden. Bei größerer Dateimenge können die Daten nur per mobilen Datenträger ins Archiv überführt werden. Wählen Sie hierfür die zweite Option "Datenträger" aus und nehmen Sie im nächsten Schritt Kontakt zu uns auf.

OPTION 1: Hochladen:

OPTION 2: Datenträger:

Wählen Sie

Copyright © 2017 FH Potsdam. Alle Rechte vorbehalten. FAQ • Kontakt • Impressum

### Schritt 2: Datei wählen, Metadaten eingeben und bestätigen

Startseite → Hochladen → Archivpaket hochladen

## Archivpaket hochladen

Hochladen Herunterladen Profil

Laden Sie ihr Transferpaket in einer ZIP-Datei hoch. Das verpackte Archivformat darf nur TIFF-Dateien enthalten. Beschreiben Sie kurz den Inhalt der Datenlieferung.  
 Das Hochladen kann einige Zeit dauern. Bitte verlassen Sie nicht diese Webseite während der Browser noch arbeitet.


Durchsuchen...

**Beschreibung der Datenlieferung:**  
 Eine Testdatei mit drei leeren Textdokumenten.

Hochladen

Die Datei wird dem Archivpaket hinzugefügt. Im Gesamtbestand wird das Transferpaket mit dem Status „Hochgeladen“ ausgegeben.

Willkommen! Sie sind angemeldet als Museum XYZ Abmelden



Startseite → Herunterladen → Bestand anzeigen

## Bestand anzeigen

	Hochladen	Herunterladen	Profil
--	-----------	---------------	--------

Projekt: **Testprojekt 01**

- o testdokument.zip 358 Bytes >> Hochgeladen << hochgeladen am 2017-02-23 um 21:28:35 Uhr
- o metadaten.xml
- o submission\_manifest.txt

1 Transferpaket(e) | 358 Bytes

Anzeige des Gesamtbestandes für den angemeldeten Produzenten

Durch einen Mausklick auf den Projekttitel oder auf die Datenlieferung kann eine detaillierte Ansicht aufgerufen werden.

### Suchergebnis: Projekt/Archivpaket

Startseite → Herunterladen → Paketsuche → Suchergebnis

## Suchergebnis

	Hochladen	Herunterladen	Profil
--	-----------	---------------	--------

Projekt:  
**Testprojekt 01**

Jahr: 2017  
ISIL: DE-XYZ-1234 (Museum XYZ)  
Finanzierung: Internes Projekt  
Rechte: -  
Projektkurator: -

**Beschreibung:** Das erste Testprojekt.

Projekthinhalte

- o testdokument.zip [transfer\_1] 358 Bytes >> Hochgeladen << hochgeladen am 2017-02-23 um 21:28:35 Uhr
- o metadaten.xml
- o submission\_manifest.txt

1 Transferpaket(e) | 358 Bytes

### Suchergebnis: Datenlieferung

Startseite → Herunterladen → Paketsuche → Suchergebnis

## Suchergebnis

	Hochladen	Herunterladen	Profil
--	-----------	---------------	--------

Transferpaket:  
**testdokument.zip**

Dateityp: zip    Server-ID: transfer\_1    Dateigröße: 358 Bytes    Dateimenge: 3 Datei(en)

**Beschreibung:** Eine Testdatei mit drei leeren Textdokumenten.

ISIL: DE-XYZ-1234 (Museum XYZ)  
Projekt: Testprojekt 01

Status  
Hochgeladen

**Erläuterung:** Dieses Transferpaket erfüllt alle Vorverarbeitungskriterien und wird in kürze an das Archivsystem übergeben

[Herunterladen](#)

Herunterladen nicht möglich, solange dieses Transferpaket nicht archiviert wurde (Status: A/P).

## ANHANG H: Datenlieferungen manuell hinzufügen

Pakete, die sich im Verzeichnis „datentraeger“ befinden, können einem Projekt manuell hinzugefügt werden. Das Verzeichnis wird zunächst auf vorhandene Daten untersucht.

Keine Daten im Verzeichnis „datentraeger“:

Name	Änderungsdatum	Typ	Größe
datentraeger_hinweis.txt	22.01.2017 17:58	Textdokument	1 KB

Ausgabe im Demonstrator:

Willkommen! Sie sind angemeldet als Administrator Abmelden

Startseite → Hochladen → Archivpaket manuell zuordnen

**Archivpaket manuell zuordnen** Hochladen Suche Profil Verwaltung

Wählen Sie eine Datenlieferung aus:

**Keine Transferpakete im Verzeichnis "Datentraeger" vorhanden.**

Daten im Verzeichnis „datentraeger“ vorhanden:

Name	Änderungsdatum	Typ	Größe
datentraeger_hinweis.txt	22.01.2017 17:58	Textdokument	1 KB
testdokument2.zip	23.02.2017 21:25	WinRAR-ZIP-Archiv	1 KB
testdokument3.zip	23.02.2017 21:25	WinRAR-ZIP-Archiv	1 KB

Ausgabe im Demonstrator:

Startseite → Hochladen → Archivpaket manuell zuordnen

**Archivpaket manuell zuordnen** Hochladen Suche Profil Verwaltung

Wählen Sie eine Datenlieferung aus:

- testdokument2.zip
- testdokument3.zip

Nach der Auswahl einer Datenlieferung folgen die Institutionsauswahl, die Projektauswahl und die Metadateneingabe.

## ANHANG I: Simulation des Archivierungsprozesses

Datenlieferungen, die Projekten hinzugefügt wurden, erhalten in der Regel den Status „Hochgeladen“. Die Administration kann diese Pakete suchen.

Erweiterte Suche (Eingabe des Status):

Startseite → Suche → Paketsuche → Paketsuche (erweitert)

**Paketsuche (erweitert)**
Hochladen
Suche
Profil
Verwaltung

Erweiterte Suche

**Institution:**

**Suche beschränken auf:**  
 Alles:  Projekte:  Archivpakete:

**Paket:**

**Status:**


**Projekt:**

**Projekt- und Datenlieferungsbeschreibung (Suchwort):**

**Jahr:**

Einfache Suche und Ergebnisse:

Willkommen! Sie sind angemeldet als Administrator
Abmelden



Startseite → Suche → Paketsuche

**Paketsuche**
Hochladen
Suche
Profil
Verwaltung

Einfache Suche

Suchen Erweiterte Suche

Treffer: 2

**Archivpakete (Projekte)**

- o Testprojekt 01 [SIL: DE-XYZ-1234] erstellt 2017

**Datenlieferungen**

- o testdokument.zip >> Hochgeladen << [ID: transfer\_1 | SIL: DE-XYZ-1234] hochgeladen am 2017-02-23 um 21:28:35 Uhr

Über das Suchergebnis kann der Archivierungsprozess initialisiert werden. Statt des Buttons „Herunterladen“ (vgl. Anhang G) wird der Administration der Schalter „Paket archivieren“ angezeigt.

## Suchergebnis aus Sicht des Administrators:

Startseite → Suche → Paketsuche → Suchergebnis

### Suchergebnis

[Hochladen](#)
[Suche](#)
[Profil](#)
[Verwaltung](#)

Transferpaket:

#### testdokument.zip

Dateityp: zip    Server-ID: transfer\_1    Dateigröße: 358 Bytes    Dateimenge: 3 Datei(en)

Beschreibung: Eine Testdatei mit drei leeren Textdokumenten.

ISIL: DE-XYZ-1234 (Museum XYZ)

Projekt: Testprojekt 01

#### Status

Hochgeladen

Erläuterung: Dieses Transferpaket erfüllt alle Vorverarbeitungskriterien und wird in kürze an das Archivsystem übergeben

[Paket archivieren](#)

## Ansicht nach aktivieren des Buttons „Paket archivieren“:

Startseite → Hochladen → ARCHIVSYSTEM: SIMULATION

### ARCHIVSYSTEM: SIMULATION

[Hochladen](#)
[Suche](#)
[Profil](#)
[Verwaltung](#)

Transferpaket erfolgreich in das INPUT-Verzeichnis transferiert.

#### INPUT

- o metadaten.xml
- o submission\_manifest.txt
- o testdokument\_1.txt
- o testdokument\_2.txt
- o testdokument\_3.txt

#### OUTPUT

Keine Dateien im OUTPUT-Verzeichnis vorhanden.

[Simulation starten](#)

PHASE 1 von 2: Transferpaket wird theoretisch vom Archivsystem abgeholt, verarbeitet und in den OUTPUT zurückgegeben

Die Datenlieferung und die Metadaten-dateien wurden in den Input-Ordner verschoben und das Transferpaket wurde entpackt. Die Simulation kann nun gestartet werden.

## Phase 1 wurde abgeschlossen:

Startseite → Hochladen → ARCHIVSYSTEM: SIMULATION

### ARCHIVSYSTEM: SIMULATION

[Hochladen](#)
[Suche](#)
[Profil](#)
[Verwaltung](#)

Transferpaket erfolgreich an das Archivsystem geliefert und verpackt zurückerhalten.

#### INPUT

Keine Dateien im INPUT-Verzeichnis vorhanden.

#### OUTPUT

- o transfer\_1\_aip.zip

[Simul. fortsetzen](#)

PHASE 2 von 2: Transferpaket wird dem Produzenten wieder zugeordnet und zur Verfügung gestellt

Die Simulation wurde fortgesetzt und Phase 2 abgeschlossen:

Startseite → Hochladen → ARCHIVSYSTEM: SIMULATION

## ARCHIVSYSTEM: SIMULATION

Hochladen Suche Profil Verwaltung

Transferpaket erfolgreich in das Produzenten-Verzeichnis transferiert.

INPUT  
Keine Dateien im INPUT-Verzeichnis vorhanden.

OUTPUT  
Keine Dateien im OUTPUT-Verzeichnis vorhanden.

Die Datenlieferung steht nun mit dem Status „AIP“ zur Verfügung und kann heruntergeladen werden.

Startseite → Suche → Paketsuche → Suchergebnis

## Suchergebnis

Transferpaket:  
**testdokument.zip**

Dateityp: zip Server-ID: transfer\_1 Dateigröße: 358 Bytes Dateimenge: 3 Datei(en)

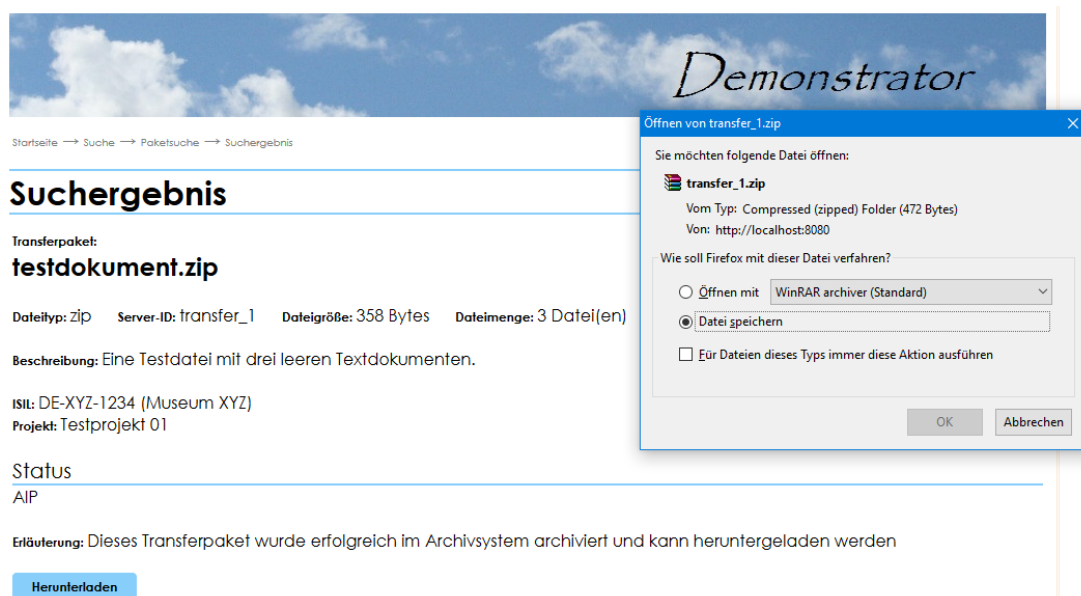
Beschreibung: Eine Testdatei mit drei leeren Textdokumenten.

ISI: DE-XYZ-1234 (Museum XYZ)  
Projekt: Testprojekt 01

Status  
AIP

Erläuterung: Dieses Transferpaket wurde erfolgreich im Archivsystem archiviert und kann heruntergeladen werden

[Herunterladen](#)



The screenshot shows a web application interface with a search result for 'testdokument.zip'. The file is in 'AIP' status and has a 'Herunterladen' button. A Firefox dialog box is overlaid on the page, asking how to handle the file 'transfer\_1.zip' (472 Bytes, Compressed (zipped) Folder). The dialog offers options to 'Öffnen mit WinRAR archiver (Standard)', 'Datei speichern' (selected), or 'Für Dateien dieses Typs immer diese Aktion ausführen'. The 'Datei speichern' option is selected, and the 'OK' button is visible.

Suchergebnis: Datenlieferung mit dem Status AIP und funktionsfähigen Button zum Download



## **Eidesstattliche Erklärung**

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus fremden Quellen entnommen wurden, sind als solche gekennzeichnet. Die Arbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt oder an anderer Stelle veröffentlicht.

---

Eric Neudhöfer | Potsdam, 27.02.2017