

FACHHOCHSCHULE POTSDAM

Fachbereich Informationswissenschaften

Studiengang Bibliotheksmanagement

Prof. Dr. Stephan Büttner

Prof. Dr. Heike Neuroth

Bachelorarbeit

**Das Potential von Open- Source- Software für
Bibliotheken in ausgewählten Ländern**

vorgelegt von:

Student: Volker- Simon Best
Studiengang: Bibliotheksmanagement
Fachsemester: 9
Matrikelnummer: s11199

Berlin, den 01.02.2015

Inhalt

1	EINLEITUNG.....	1
2	FREI, OFFEN ODER PROPRIETÄR – ENTWICKLUNG, DEFINITION UND VERBREITUNG.....	4
2.1	DIE VERBREITUNG VON OPEN- SOURCE- SOFTWARE.....	9
2.2	BIBLIOTHEKSSPEZIFISCHE OPEN- SOURCE- SOFTWARE IN DEN USA UND DEUTSCHLAND.....	13
3	PROBLEMATIK I - URHEBERRECHT, PATENTRECHT UND SOFTWARELIZENZIERUNG.....	24
3.1	URHEBERRECHT.....	24
3.2	SOFTWARELIZENZEN.....	27
3.3	PATENTRECHT UND FOSS- LIZENZEN.....	28
4	PROBLEMATIK II – POLITIK, PHILOSOPHIE UND SOZIALE NORMEN	32
4.1	POLITISCHE STATEMENTS – DIE EU UND DEUTSCHLAND.....	34
4.2	POLITISCHE STATEMENTS – USA.....	37
4.3	PHILOSOPHIEN UND NORMEN – DIE OPEN- SOURCE- COMMUNITY UND DAS BIBLIOTHEKARISCHE VERSTÄNDNIS DER INFORMATIONSDIENSTLEISTUNG.....	39
5	PROBLEMATIK III – DER CODE.....	43
5.1	BEISPIEL I: EIN SUCHINDEX AUS LOD- DATENSÄTZEN FÜR SOLR.....	43
5.2	BEISPIEL II: EINBINDUNG VON PATENTINFORMATIONEN IN EIN INSTITUTIONELLES REPOSITORYUM.....	46
6	FAZIT / AUSBLICK.....	49
7	ANHANG.....	I
8	LITERATURVERZEICHNIS.....	X

Verzeichnis der Abbildungen

Abbildung 1.....	21
Abbildung 2.....	23
Abbildung 3.....	39
Abbildung 4.....	47
Abbildung 5.....	48
Abbildung 6.....	IX

Verzeichnis der Tabellen

Tabelle1.....	5
Tabelle2.....	11
Tabelle3.....	15

Verzeichnis des Anhangs

Anhang I: Transkript.....	I
Anhang II.....	IX

Verzeichnis der Abkürzungen

API	Application Programming Interface
BIT	Bundesstelle für informationstechnik
BTS	Bug Tracking System
CTSS	Compatible Time- Sharing System
CVS	Concurrent Version System
DOI	Digital Object Identifier
Fedora	Flexible Extensible Digital Object Repository
FOSS	Free and Open Source Software
FSF	Free Software Foundation
FSFE	Free Software Foundation Europe
GOKb	Global Open Knowledge Database
GNU	Gnu is not Unix
IPR	Intellectual Property Right
IDE	Integrated Development Environment
IRC	Internet Relay Chat
LTS	Long Term Support
OLE	Open Library Environment
OSI	Open- Source- Initiative
OSS	Open- Source- Software
PERL	Practical Evaluation and Reporting Language
SaaS	Software as a Service
TCO	Total Cost of Ownership
WIPO	World Intellectual Property Organisation

1 Einleitung

„It is time for librarians to stand up for open source.“¹

Im Rahmen des neuen Paradigmas der „Openness“ rückt auch die verwendete IT in den Fokus des Diskurses. Als Prinzip stützt sich dieses Paradigma auf die Forderungen nach offenen Infrastrukturen, offenen Inhalten und offenen wissenschaftlichen Prozessen.² Dies beinhaltet auch die Verwendung von Open-Source-Software. Ging man Anfang der 2000er Jahre noch davon aus, dass sich bewährte proprietäre IT-Strukturen in Bibliotheken nicht durch Open-Source-Software (OSS) ersetzen lassen³, zeigt sich heute ein anderes Bild. Komplexe ILS-Anwendungen, diverse CMS, Kataloge, Discovery- und Retrievalsysteme, Repositorien, Cloud-Dienste oder auch größere Publikationsserver lassen sich mittlerweile mit stabiler OSS betreiben. In der Forschung und im alltäglichen Gebrauch öffentlicher Verwaltungen findet man vereinzelt die Verwendung von OSS vor, vom Zählen einzelner Zellkolonien in der Biotechnologie über die Sequenzierung von Verhaltensmustern in der psychotherapeutischen Forschung bis zur Administration komplexer Netzwerke in Stadtverwaltungen lassen sich auf OSS basierende, einzelne Module oder komplette Systeme finden. Im Rahmen der Umsetzung lassen sich vier Problemfelder identifizieren: juristische, finanzielle, politische und philosophische. Wie Morgan bemerkt, spielen soziale Aspekte eine wichtige Rolle wenn es um die Implementierung von OSS in proprietäre Strukturen geht: „[...] any computing problem can roughly be divided into 20 percent technology issues and 80 percent people issues.“⁴ Wie die meisten am Thema beteiligten kommt er zu dem Schluss dass OSS sowohl Philosophie als auch Prozess ist. Daneben sind Kürzung von Personal und knappe Erwerbungsbudgets seit längerem bekannte Probleme von Bibliotheken und haben sich auch aufgrund hoher Lizenzkosten für unflexible IT-Strukturen verschärft. Es ist allerdings ein Irrtum zu glauben dass bei der Verwendung von OSS keine Kosten entstehen. Dass eine Umstellung auf eine offene Struktur nicht ausschließlich ökonomische Gründe hat soll in dieser Arbeit gezeigt werden, denn wie

¹ Vgl. JAFFE, Lee David; CAREAGA, Greg. Standing up for open source. *Library Philosophy and Practice*, 2007, 9. Jg., Nr. 2, S. 21.

² Vgl. LOHMEIER, Felix; MITTELBACH, Jens, 2014, Offenheit statt Bündniszwang. *Zeitschrift für Bibliothekswesen und Bibliographie*. 2014. Jg. 61, Nr. 4-5, S. 209-214.

³ Vgl.: BREEDING, Marshall. The open source ILS: still only a distant possibility. *Information Technology and Libraries*, 2002, 21. Jg., Nr. 1, S. 16.

⁴ Vgl.: MORGAN, Eric Lease. Possibilities for open source software in libraries. *Information Technology and Libraries*, 2002, 21. Jg., Nr. 1, S. 12.

Grassmuck schon 2002 bemerkt: „Das Phänomen der freien Software erinnert dabei auch an eine in den Hintergrund getretene Tatsache: Die Informationsgesellschaft ist keine alleinige Erfindung der Industrie, und ihre Dynamik verdankt sie nicht nur den Kräften des Marktes.“⁵

Die Entwickler- Communities freier Software sind heute größer und besser vernetzt als dies noch in den 80er Jahren der Fall war. Ein Grund warum die Community nach der Gründung von Software- Unternehmen wie Microsoft 1975 enger zusammenrückte war sicherlich die Beanspruchung auf das intellektuelle Copyright und Kommerzialisierung des UNIX- Systems seitens AT & T mit der Folge der Gründung des GNU- Projekts 1983 und der Free Software Foundation am MIT 1985. Heute gibt es dank riesiger, heterogener Communities immer mehr Anwendungen die mit Long- Term- Support (LTS) erscheinen, nicht selten gelten dabei Zeiträume von 5 Jahren⁶, was den Einsatz durchaus planbar macht. Offen entwickelte Quellcodes erreichen häufiger eine höhere Qualität und geringere Fehlerquote als proprietäre Codes aus Unternehmensprojekten und in Arbeitsfeldern die weiterhin proprietär dominiert sind.⁷ Die Usability hat sich ebenfalls gebessert und auch Nicht- Informatikern ist es möglich OSS zu nutzen und für verschiedene Bedürfnisse zu konfigurieren. OSS weist also viele Charakteristika eines ernst zu nehmenden Konkurrenten für etablierte Software- Anbieter auf. Was bedeutet dies für Bibliotheken?

Der Innovationsdruck auf Bibliotheken ist mit der digitalen Transformation sehr groß geworden. Im Zuge der Open- Access- und Open- Science- Bewegungen lässt sich die Forderung nach der Offenheit und Verfügbarkeit aller gesammelten und erschlossenen (Meta-) Daten und Information als primäres Handlungsziel identifizieren, was kein genuin neues Kompetenzfeld darstellt. Wie Mittelbach bemerkt bremsst aber die jahrzehntelange Abhängigkeit von konventionellen IT- Unternehmen und die damit verbundene Schaffung von „innovationsfeindlichen Kartellstrukturen“ das Innovationspotential von Bibliotheken gezielt aus.⁸ Lassen sich die proprietären Systeme langfristig ersetzen? Welche OSS- Anwendungen machen für welche Bibliotheken Sinn? Mit einer Einschätzung des Potentials von OSS soll gezeigt werden wie nah dieses Thema an

⁵ Vgl. GRASSMUCK, Volker; SIBER, Karl Heinz. *Freie Software: Zwischen Privat- und Gemeineigentum*. Bundeszentrale für politische Bildung, 2004, S.10.

⁶ Dies ist etwa bei diversen freien Betriebssystemen wie den Ubuntu- Derivaten die Norm. Einen ständig aktualisierten Überblick über die verschiedensten Ausprägungen freier GNU/Linux- Betriebssysteme erhält man unter: <http://distrowatch.com/>

⁷ Vgl. HEISE ONLINE, 2014, Studie zur Softwarequalität: *Open Source schlägt proprietär*. [online]. 2014. [Abgerufen am 16 November 2015]. Verfügbar unter: <http://www.heise.de/newsticker/meldung/Studie-zur-Softwarequalitaet-Open-Source-schlaegt-proprietaeer-2175954.html>

⁸ Vgl. MITTELBACH, Jens. *Modernes Datenmanagement: Linked Open Data und die offene Bibliothek*. o-bib. Das offene Bibliotheksjournal/herausgegeben vom VDB, 2015, 2. Jg., Nr. 2, S. 61-73.

der ursprünglichen, neuzeitlichen bibliothekarischen Philosophie, nämlich der Beförderung der freien Forschung und des Teilens von Wissens sein kann. Es soll weiterhin dargestellt werden wie sehr das persönliche Engagement und die IT- Kompetenz des einzelnen Bibliothekars an die Frage nach dem Potential von OSS gekoppelt ist, denn wie Engard ausführt: „Indeed one could say the primary function of a librarian is to facilitate the transfer of knowledge, thus empowering their communities, which is the same role free software plays.“⁹

In dieser Arbeit wird die Anwendung von Open- Source in Deutschland und den USA betrachtet. Dies liegt in der Tendenz des deutschen Bibliothekswesens begründet innovative Konzepte der Informationstechnik vorrangig aus dem amerikanischen Raum zu adaptieren.¹⁰ Es gibt aber auch viele andere Länder die äußerst erfolgreich nationale Strategien und Open- Source- Implementierungsmodelle für Bibliotheken erstellt und umgesetzt haben wie Dänemark, Indien, oder einige afrikanische Länder.¹¹ Aufgrund des begrenzten Rahmens der Bachelorarbeit wurde nur ein Interview über den Einsatz von Open- Source- Software in deutschen Bibliotheken durchgeführt und auf weitere interviews mit amerikanischen Bibliothekaren verzichtet. Als äußeren Bezugsrahmen orientiert sich die Arbeit an dem Regulations- Modell¹² von Lawrence Lessig, einer der Initiatoren des Creative- Commons- Lizenzmodells, nach dem auf den Menschen und seine Lebenswelt vier Regel bildende Entitäten einwirken die aus dem Recht, den sozialen Normen und dem Markt bestehen, wobei mit der Transformation in die Informations- oder Wissensgesellschaft schließlich die Software und der ihr zugrunde liegende Code als Entität hinzu gekommen sind.¹³ Lessig beschreibt die Technik an sich, das Internet und die Software als eine eigenständige, normative Kraft. Dies ist neben den Überlegungen Richard Stallmanns in den 80er Jahren das erste Mal dass Quellcode über die technische Wirkung hinaus gesellschafts- veränderndes Potential zugesprochen wird („Code is Law“). Freie Software generiert nach Lessig einen neuen Blick auf kollaborative Nutzung von Technik und ändert die rechtliche Situation des geistigen Eigentums.

⁹ ENGARD, Nicole. *Practical open source software for libraries*. Elsevier, 2010.

¹⁰ Vgl. ALTENHÖNER, Reinhard. Deutschland und die USA: Aspekte der Informationsinfrastruktur im Wechselspiel von Orientierung und Abgrenzung. *Bibliothek Forschung und Praxis*, 2015, 39. Jg., Nr. 2, S. 150-157.

¹¹ Zu nennen sind hier das TING- Projekt dänischer Bibliotheken oder die Initiative Digital India die in beiden Ländern zu einer breiten Adaption von OSS im öffentlichen Sektor geführt haben. In Afrika gibt es seit 2005 vermehrt Installationen von Open- Source- BMS/ILS.

¹² Eine kurze aber vollständige Darstellung des Modells findet sich bei: LUTTERBECK, Bernd. Die Zukunft der Wissensgesellschaft. *Lutterbeck, Bernd et al., Open Source Jahrbuch*, 2006, S. 445-465.

¹³ Vgl. LESSIG, Lawrence. *Code and other laws of cyberspace*. Basic Books, New York 1999.

Diese Arbeit folgt der Definition von Quelltext oder Code selbst als: „[...] the executable specification of a software system’s behaviour.“¹⁴ Quelltext kontrolliert das Verhalten eines Programms und der Hardware, zum Nutzer und zu anderen Programmen. Eingeschlossen in den Begriff ist meist auch die Bündelung (packages/modules) von Dateien und Dokumenten die den Quelltext konstituieren. Diese wiederum enthalten mit der Syntax und Semantik der verwendeten Programmiersprachen strukturierte und unstrukturierte Daten, die auch als „linguistische“ Daten bezeichnet werden wie Kommentare der Entwickler, String- Literale, Identifier- Namen, sowie Namen von lokalen oder globalen Variablen. Das bekannteste und am meisten verbreitete OSS- Projekt, der Linux- Kernel, besteht aktuell aus 21.198.783 Zeilen Quellcode und 65.177 Dateien.¹⁵ Bei Installationen werden allerdings nur die für die jeweilige Hardware nötigen Zeilen kompiliert. Die große Stärke des Linux- Kernels, nämlich in verschiedenste, herstellerunabhängige Hardware integriert werden zu können, drückt sich auch darin aus dass der größte Teil des Codes entsprechende Treiber beinhaltet. In Kapitel 5 werden Beispiele für Quelltexte von Bibliothekssoftware und dessen Bearbeitung näher erläutert.

2 Frei, offen oder proprietär – Entwicklung, Definition und Verbreitung

Man muss sich darüber klar sein dass am Beginn der Softwareentwicklung, im universitären aber auch im privatwirtschaftlichen Sektor, das Teilen und die Offenheit von Software alltäglich waren¹⁶ und die zunehmenden Einschränkungen erst mit der Kommerzialisierung bestimmter (Teil-)Systeme überhand nahmen, so dass bis heute von Kartellstrukturen gesprochen werden kann. Software war bis Ende der 60er Jahre eine Ergänzung der Hardware und keine eigenständige oder, um mit Lessig zu sprechen, normende Entität. Es gab bis zu diesem Zeitpunkt also keine rechtliche Beschränkung hinsichtlich des Kopieren, Änderns oder Verteilung von Quellcodes, sondern die un-

¹⁴ Vgl. THOMAS, Stephen W.; HASSAN, Ahmed E.; BLOSTEIN, Dorothea. Mining Unstructured Software Repositories. In: *Evolving Software Systems*. Springer Berlin Heidelberg, 2014. S. 139-162, S. 141.

¹⁵ Dies ergab eine Terminal- Abfrage nach Klonen des git- Repositoriums für den Linux- Kernel auf den eigenen Rechner am 26.01.2016. Git ist ein freies Programm zur verteilten Versionsverwaltung von Dateien. Über reguläre Ausdrücke lassen sich diese nach bestimmten Variablen untersuchen.

¹⁶ Noch bis in die 60er Jahre gab es keine Unterteilung in Soft- und Hardwareproduzenten. Software wurde wie eine „Betriebsanleitung“ zusammen mit der Hardware an Nutzer ausgeliefert die meist selbst Programmierer oder Entwickler waren. Der Code wurde also selbst von Marktführern wie IBM oder Hewlett Packard als offen und veränderbar geliefert.

ausgesprochene Ethik der „kollektiven Invention“.¹⁷ Dies gilt als die erste Phase in der Entwicklungsgeschichte von Software (1940-1970). Die zweite Phase mit vermehrt urheberrechtlich geschützter Software, Firmengründungen (Microsoft 1975, Apple 1976) und der parallelen Etablierung freier Programmierer Subkulturen wird in den Zeitraum bis 1980/85 verortet. Bis 2000 gilt proprietäre Software als maßgeblicher Innovationsmotor in der dritten Phase. Die Lizenzierung und die Kosten proprietärer Software wurden aber für viele Start- Ups oder mittlere Unternehmen untragbar so dass die Jahre von 2000 bis heute als der Zeitraum der Ausdifferenzierung der Open-Source- Dynamik gilt und Innovationen aus diesem Feld generiert werden und eine zunehmende Partizipation beider Seiten der Softwareentwicklung erfordert. Im folgenden Kapitel werden genauere Zahlen zur zunehmenden Verwendung von Open – Source genannt.

Der urheberrechtliche Schutz von Programmen wurde lange als unnötig abgetan, da die Hardware ausreichend patentiert wurde. Bis zum Jahr 1978 gab es in Amerika lediglich 1200 eingetragene Softwarepatente.¹⁸ 1978 war das Jahr in dem schließlich Software als patentfähig ohne den Zusatz „rule of doubt“ (unter Vorbehalt) eingestuft wurde. Ab diesem Zeitpunkt lässt sich von eigentlich proprietärer Software sprechen. Nachfolgend eine tabellarische Auflistung¹⁹ der für die IT wichtigsten Software- Neuerungen zwischen 1979 und 1999, also dem Zeitraum in dem vor allem das proprietäre Modell erfolgreich und etabliert war.

JAHR	Proprietär	Frei/Open- Source
1979	Oracle Database	
1981	MS-DOS	
1982	Norton Utilities	
1983	Microsoft Word, IBM DB2, Lotus 1-2-3	GNU project
1984	MacOS, MacPaint, MacWrite, MacDraft, AppleWorks, Quicken	LaTeX
1985	StarOffice, PageMaker, Windows	
1987	Illustrator, OS/2	GCC, PERL
1989	NeXTSTEP	BSD und MIT- Lizenzen, POST-

¹⁷ Vgl. SCHRAPE, Jan-Felix. *Open Source Softwareprojekte zwischen Passion und Kalkül*. Stuttgarter Beiträge zur Organisations- und Innovationsforschung, SOI Discussion Paper, 2015.

¹⁸ Vgl. MERGES, Robert P., et al. *Software & Internet Law*. Aspen Publishers, 2011.

¹⁹ Die Daten stammen aus einem Blogbeitrag von Allison Randal, einer bekannten Persönlichkeit der freien Entwickler- Community und Perl / C- Programmiererin. Sie hatte mehrmals Führungsposition in der Perl Foundation und Python Software Foundation inne und ist seit Mai 2015 Präsidentin der Open- Source- Initiative. Vgl. Allison Randal, 2015, The Future of Open Source. [online]. 2015. [Abgerufen 23 Januar 2016]. Verfügbar unter: <http://allisonrandal.com/2015/07/22/the-future-of-open-source/>

		GRES, Cygnus- Support
1990	Photoshop	CERN http
1991		Linux- Kernel, PYTHON
1993	Mosaic Web Browser	Debian, FreeBSD, NetBSD, Lua
1994	Netscape Navigator	RedHat
1995	Amazon.com, Internet Explorer, Sun Java, JavaScript,	MySQL, Apache HTTP Server, PHP, Ruby, PostgreSQL
1996	Flash	GIMP, KDE
1997	Google Search, Netscape Communicator	
1998		Mozilla Application Suite
1999	Mac OS X Server	GNOME, Asterisk, CUPS

Tab. 1: Wichtige Software- Neuerungen 1979- 1999
Quelle: eigene Darstellung in Anlehnung an Randal 2015

Die Aufteilung in proprietär und offen wurde zwar erst in den 1980er Jahren endgültig verfestigt, aber mit der Gründung von Microsoft 1975 gab es einen der prägendsten Momente die den weiteren Diskurs bestimmte. Der von Bill Gates geschriebene, offene Brief „An Open Letter to fellow Hobbyists“²⁰ beschuldigte die freie Entwicklergemeinschaft 1976 des Diebstahls in Bezug auf sein Programm Altair BASIC und stellte die Behauptung auf gute Software lasse sich nur gegen direkte Bezahlung entwickeln und sollte nicht geteilt werden. Gates selbst schrieb aber seine erste proprietäre Software mithilfe der öffentlich finanzierten IT- Infrastruktur und baute auf frei geteiltem Code auf. Das erste MS-DOS Betriebssystem, das Microsoft zu großen Teilen von einer anderen Firma kaufte, war so erfolgreich weil IBM sich verpflichtet hatte seine Hardware nur mit diesem System auszuliefern, also nicht weil das Programm zu diesem Zeitpunkt das beste auf dem Markt gewesen wäre.²¹ Richard Stallmann, der Gründer der Free Software Foundation, weist unermüdlich auf die Unterschiede von proprietärer und freier oder Open- Source- Software hin. Die gestiegene Kommerzialisierung der GNU/Linux- Derivate im Sektor der Betriebssysteme, vor allem die Ubuntu- Serie von Firmen wie Canonical und deren Kooperation mit E-Bay und Amazon, sieht er als tendenziell desaströs für die Mündigkeit der User und das Offenheitsprinzip, obwohl der verwendete Kernel und die meisten Anwendungen weiterhin den Open- Source Leitlinien entsprechen.²² Die Definition von Software die dem Attribut frei ent-

²⁰ Vgl. Web.archive.org, 2016, Computer Notes Magazine volume 1 issue 9 at the START-UP Gallery. [online]. 2016. [Aufgerufen 10 Januar 2016]. Verfügbar unter: https://web.archive.org/web/20120323162213/http://startup.nmnaturalhistory.org/gallery/notesViewer.php?ii=76_2&p=3

²¹ Grassmuck 2004

²² Vgl. Torvalds, Stallman., 2015, Stallman: If you want freedom don't follow Linus Torvalds. *Computerworld* [online]. 2015. [Accessed 22 December 2015]. Available from: http://www.computerworld.com.au/article/195096/stallman_want_freedom_don_t_follow_linus_torvalds/

spricht hat nach der seit 1986 gültigen Fassung vier Charakteristika: „A program is Free Software if you, the user, have the four essential freedoms. Freedom 0 is the freedom to run the program as you wish. Freedom 1 is the freedom to study the source code and then change it to make the program do what you wish. Freedom 2 is the freedom to help your neighbour – that's the freedom to make and distribute exact copies of the program to others, when you wish. And Freedom 3 is the freedom to contribute to your community: that's the freedom to distribute copies of your modified versions, when you wish.“²³ Seit 1998 gibt es die Definition der Open- Source- Initiative, die den Begriff freier Software mit Open- Source- Software ersetzt, die 10 Punkte umfasst, sowie die Vergabe eines Zertifikats für Software die diesen Punkten entspricht eingeführt hat. Das Siegel ist dabei nur auf die Software anzuwenden und kann nicht auf die Lizenz übertragen werden, was sicherstellt das es sich bei einem entsprechenden Softwarepaket um eine Open- Source- Distribution handelt. Die zehn Punkte, die mehr einem Standard als einer Lizenz entsprechen, umfassen: 1. Freie Weiterverteilung (keine Lizenzgebühr/Software darf als Bestandteil mit Software aus unterschiedlichen Quellen verteilt werden), 2. Quellcode (Das Programm muss als Source- Code und in binärer Form frei verfügbar sein, wird dieser nicht weitergegeben muss ein Hinweis auf den Bezug zum Selbstkostenpreis vorhanden sein). 3. Abgeleitete Versionen (erlaubt modifizierte Versionen der ursprünglichen Version und die Veröffentlichung derselben unter den gleichen Bedingungen), 4. Integrität des Quellcodes (Einschränkungen die Verbreitung abgeleiteter Versionen: keine Verbreitung mit Patches die das Programm bei Kompilierung verändern, Software die aus der Abänderung des ursprünglichen Codes besteht muss einen anderen Namen oder Versionsnummer aufweisen, ansonsten darf die Lizenz die Verbreitung geänderten Codes nicht untersagen), 5. Keine Diskriminierung einzelner Personen oder Gruppen (die Software ist für alle zur Verfügung zu stellen, unabhängig von Herkunft, Geschlecht oder sonstigen Unterschieden), 6. Keine Diskriminierung kommerzieller Nutzer (die kommerzielle Nutzung ist für alle Anwendungsbereiche zu gestatten), 7. Lizenzerteilung (Lizenz für alle Nutzer bei Erhalt anwendbar ohne dass dieser eine Registrierung durchführen oder weitere Lizenzen erwerben muss), 8. Lizenzneutralität (Lizenz betrifft nur den Quellcode und keine spezifische Distribution), 9. Lizenzneutralität (die Verbreitung in Verbindung mit anderer, auch proprietärer Software ohne eunsichtbaren Code ist restriktionsfrei zu gestatten) 10. Die Lizenz muss technologie- neutral sein (die Verbreitung darf nicht auf

²³ Vgl.: Gnu.org, 2016, What is free software?- GNU Project - Free Software Foundation. [online]. 2016. [Abgerufen 8 Januar 2016]. Verfügbar unter: <http://www.gnu.org/philosophy/free-sw.html>

eine bestimmte Technologie oder ein bestimmtes Interface beschränkt sein).²⁴ Diese nicht in allen Punkten mit der Philosophie der durch die FSF geprägten Definition übereinstimmende Erweiterung, wurde von den OSI-Initiatoren als eine Art Image-Kampagne und Hinwendung zu möglichen Investoren verstanden, nicht zuletzt mit dem Argument Stallmanns Definition von „frei“ sei zu kommunistisch und ungeeignet für die neue Generation von Softwareentwicklern die sich auf dem freien Markt behaupten wollten und für die die Leistungs- und Vermarktungsfähigkeit der Programme wichtiger waren als eine politische Stellungnahme. Dabei hat Stallman selbst bereits früh auf mögliche Geschäftsmodelle hingewiesen und auch vorgelebt, indem er auf entsprechenden Trägermedien GNU-Software und Handbücher verkaufte. Die Einnahmen generierte er also nie aus der Software selbst sondern deren Distribution, dem Service und Spenden, einem Modell das sich bis heute in der FSF bewährt.²⁵ Er sieht in der Umdeutung des Begriffs durch die OSI eine Hinwendung zum reinen Pragmatismus und eine Konzentration auf den technischen und kommerziellen Aspekt was die Freiheit und die Community vernachlässige. Es wurde zu Recht darauf hingewiesen dass Quelloffenheit per se keine Veränderbarkeit des Codes sicherstellt und der Begriff ein Vehikel für Softwarelizenzierungen und Patentierungen unfreier Art darstellen kann. So führen mehrere Softwareprodukte diverser Unternehmen erfolgreich das Siegel Open-Source, erlauben aber keine eigentliche Modifikation ihres Quellcodes.²⁶ In den Medien und in der IT-Community hat sich der Begriff Open-Source durchgesetzt und wird synonym mit freier Software verwendet, was politisch bedenklich bleibt. Weiter differenziert wird zwischen Commercial- und Community-Open-Source-Software. Erstere wird innerhalb einer privatrechtlichen Körperschaft mit dem Ziel entwickelt, letztere innerhalb einer non-profit Gemeinschaft. Verkürzt könnte man sagen dass die Bezeichnung Freie Software eher dem Ethos einer politischen Bewegung entspricht und der Begriff Open-Source-Software ein reines Entwicklungsmodell darstellt, beide aber Parallelen mit einer großen Schnittmenge sind, was sich auch in dem oft verwendeten Kürzel Free/Libre Open Source Software (FOSS/FLOSS) ausdrückt. Als Effekt in der Geschäftswelt hat sich durch den Begriff Open-Source die Bereitschaft von Firmen und Konzernen erhöht an Software-Projekten mit freien Communi-

²⁴ Siehe unter: opensource.org, 2016, The Open Source Definition | Open Source Initiative. [online]. 2016. [Abgerufen 9 Januar 2016]. Verfügbar unter: <http://opensource.org/docs/osd>

²⁵ Eine kurze Darstellung von Stallmanns Motiven und eine ausführliche Beschreibung der in der Softwareentwicklung üblichen Geschäftsmodelle bietet Buxmann, vgl. BUXMANN, Peter; DIEFENBACH, Heiner; HESS, Thomas. *Die Softwareindustrie: Ökonomische Prinzipien, Strategien, Perspektiven*. Springer-Verlag, 2015.

²⁶ Grassmuck 2004

ties und entsprechender Software zu arbeiten. OSS ist damit auch kommerzielle Software.

Abzugrenzen sind die Begriffe Freeware, Shareware und Public- Domain- Software. Freeware erlaubt weder Einsicht noch Modifikation des Quellcodes aber eine beliebige Verteilung. Sie ist in der Regel lizenzkostenfrei. Für Shareware gelten bis auf die Lizenzkosten diesselben Bedingungen. Es entstehen Lizenzkosten pro Installation und es gibt meist eine kostenfreie aber begrenzte Testphase. Public Domain Software ist eine zum Gemeingut gewordene Software und damit, inklusive des Quellcodes kostenfrei ohne lizenzrechtliche Einschränkungen verfügbar.

2.1 Die Verbreitung von Open- Source- Software

Da es mittlerweile für fast jede proprietäre Applikation eine oder mehrere Komplementäre in der Open- Source- Welt gibt sei hier nur auf die wichtigsten und für die grundlegende IT- Infrastruktur als Basis dienende Anwendungen verwiesen. Führt man sich vor Augen dass sowohl Google als auch Facebook, wie unzählige weitere Big-Data- Firmen ihre riesigen Serverkomplexe auf einer GNU/Linux- basierten Architektur laufen lassen und Firefox oder Android auf fast jedem Rechner oder Smartphone zu finden sind, ist es eine Tatsache dass jeder Nutzer des Internet tagtäglich Open-Source- Anwendungen nutzt ohne es unbedingt zu wissen, wenn diese Anwendungen auch in sonst geschlossene Systeme eingebunden sind. Quelloffene Software dominiert den Markt von Datenbank- und Content- Management- Systemen. Auf dem Markt der Webserver hält der Apache HTTP Server momentan einen Anteil von 35,59% gefolgt von Microsoft mit 26,3% und Google mit 2,26%.²⁷ Bis auf eine Ausnahme im Spätsommer 2014 ist Apache seit 1995 in diesem Segment Marktführer. Die Open Document Foundation, die sich 2010 aus der Aufspaltung des Open Office Teams gegründet hat, beziffert die Anzahl der Downloads ihres Office- Pakets namens LibreOffice auf 120 Millionen seit Entstehen 2010.²⁸ Zieht man in Betracht dass die meisten Linux- Distributionen das LibreOffice Paket in ihre Anwendungsbündelung integriert haben, ist die Anzahl der aktiven Nutzer weitaus höher.²⁹ Einer der Gründe warum

²⁷ So der Stand im Dezember 2015, vgl. News.netcraft.com, 2016, Web Server Survey | Netcraft. [online]. 2016. [Abgerufen 9 Januar 2016]. Verfügbar unter: <http://news.netcraft.com/archives/category/web-server-survey/>

²⁸ Vgl. VIGNOLI, Italo, 2016, LibreOffice: the numbers. *The Document Foundation Blog* [online]. 2016. [Abgerufen 9 Januar 2016]. Verfügbar unter: <https://blog.documentfoundation.org/blog/2016/01/01/libreoffice-the-numbers/>

²⁹ Fundierte Angaben über Anwender die ein genuin GNU/Linux- basiertes Betriebssystem mit ausschließlich freien Applikation verwenden lassen sich nicht finden aber Schätzungen gehen von min-

LibreOffice noch nicht das Monopol von Microsoft bei der Text- und Bildbearbeitung gebrochen hat sind die bei den meisten Publikationen verwendeten Formate. Um bei der Publikation und der Formulierung von wissenschaftlichen Texten eine volle Kompatibilität zu erreichen ist für Texte weiterhin das Verwenden des DOC/DOCX- Formats, für Tabellen XLS/XLSX und bestimmte Bilder oder Grafiken PPT/PPTX nötig. Nun kann man zwar mit Libre- Office und anderen Alternativen Dokumente mit diesen Formaten erstellen, bearbeiten und in alle proprietären Formate exportieren aber Microsofts Vorgehen (man könnte behaupten Geschäftsmodell) bei jeder neuen Version der Office- Suite die Formate leicht zu verändern und keine vollständige Dokumentation dazu zu veröffentlichen verursacht immer noch Kompatibilitätsprobleme und sichert somit das Monopol.³⁰ Das auf dem Linux- Kernel basierende Betriebssystem Android erreicht seit Jahren einen stabilen globalen Marktanteil von ca. 80%, mit einer Anzahl von 1,4 Millionen Aktivierungen täglich (2014).³¹ Google sorgt allerdings dafür dass in die eigentlich freie Software Werbeanzeigen eingebettet werden, die sich nach den Vorstellungen des jeweiligen Kunden und dessen Big- Data- Strategie konfigurieren lassen, was nach der Definition Stallmanns dem Kodex freier Software nicht mehr entspricht. Bis auf den Markt der Betriebssysteme für den Desktop haben sich linux- basierte Anwendungen fast überall als alternative und in genannten Beispielen als marktführende Softwareangebote etabliert. Es zeigen sich in diesem Bereich für das Jahr 2015 folgende Anteile: Windows 91%, Mac-OS: 7% und Linux 1,6%.³² Laut einer jährlichen Studie zur Zukunft von Open Source (Future of Open Source Survey der Firma Black Duck) nutzen 78% aller Firmen die IT- affin arbeiten ganz oder teilweise freie Software, 66% entwickeln selbst Software auf Open- Source Basis, 93% gaben an die Nutzung habe sich gesteigert oder sei gleich geblieben und 55% gehen davon aus dass OSS sicherer ist.³³ Die Studie geht von über 1,4 Millionen freien Software-Projekten aus. Grundsätzlich ist auf der unteren Ebenen der IT- Infrastruktur überall

destens 80 Millionen aktiven Linux- Nutzern aus, siehe unter: Linuxcounter.net, 2016, The Linux Counter Project - Statistics about Linux, its Users and more. [online]. 2016. [Abgerufen 9 Januar 2016]. Verfügbar unter: <https://www.linuxcounter.net/>

³⁰ Ein ständig aktualisierter Vergleich beider Office- Suiten findet sich unter: Wiki.documentfoundation.org, 2016, Feature Comparison: LibreOffice - Microsoft Office - The Document Foundation Wiki. [online]. 2016. [Abgerufen 10 Januar 2016]. Verfügbar unter: https://wiki.documentfoundation.org/Feature_Comparison:_LibreOffice_-_Microsoft_Office

³¹ Vgl. BLISCHKE, Johannes. *Betriebssysteme für Smartphones: Android vs. iOS*. Bachelor+ Master Publication, 2015.

³² Vgl. Netmarketshare.com, 2016, Operating system market share. [online]. 2016. [Abgerufen 11 Januar 2016]. Verfügbar unter: <http://www.netmarketshare.com/operating-system-market-share.aspx>

³³ Es wurden 1300 Firmen befragt. Die Studie geht von einer Nutzungssteigerung von OSS in amerikanischen Unternehmen von fast 50 % seit 2010 aus. Vgl. De.slideshare.net, 2016, 2015 Future of Open Source Survey Results. [online]. 2016. [Abgerufen 11 Januar 2016]. Available from: http://de.slideshare.net/North_Bridge/2015-future-of-open-source-study

OSS zu finden. Geht man auf die Ebene der betrieblichen Anwendungssoftware dominieren immer noch proprietäre Anbieter. Unternehmenssoftware umfasst die Bereiche des Customer- Relationship Management (CRM), Enterprise Resource Planning (ERP), Supply Chain Management (SCM), Business- intelligence (BI) und wenig differenziert das Feld des Dokumenten-, Wissens-, und Workflowmanagements.³⁴ Dieser Bereich hat sich aber mittlerweile im OSS- Sektor ausdifferenziert und es wird von einem stetigen Wachstum für die nächsten Jahre ausgegangen.³⁵ Das größte Repository für freie Software, sourceforge.net, bietet im Bereich der Unternehmenssoftware 15 Kategorien an, von Customer- Relationship- management (CRM) über Human Resources bis zum Wissensmanagement. Um einen Überblick über die aktuelle Größenordnung von OSS in diesem Bereich zu bekommen folgt eine Zusammenstellung der wöchentlichen Downloads der drei beliebtesten Anwendungen für jede Kategorie von „Enterprise Software“.³⁶ Bei den doppelten Einträgen handelt es sich um modularisierte Software die für verschiedene Bereiche eingesetzt werden kann.

Kategorie	1	2	3
Financial	Open Office: 939.564	GnuCash: 11.127	Dolibarr ERP-CRM: 3.933
Scheduling	OpenProj.: 6.236	Vtiger CRM: 5.267	DavMail: 2185
Office Suite	Open Office: 939.564	Scribus: 18.604	ProjectLibre: 12.714
Business	PortableApps.com: 907.280	Liferay Portal: 43.776	Pentaho: 26.049
Human Resources	OrangeHRM: 1.603	OpenKM – DMS: 1.468	Kaldin: 1.330
Time Tracking	Ganib: 3.522	Task Coach: 1.751	Collabtive: 915
To-Do Lists	FreeMind: 21.221	Ganib: 3.522	TaskCoach: 1.667
Project- management	Project Libre: 12.714	Open Proj: 6.236	FreePlane: 4.890
Medical	OpenEMR: 872	OpenMRS: 892	Ginkgo CADx: 456
E- Commerce / Shopping	cpuminer: 5.163	UniCenta POS: 2.975	Zen Cart: 1.041
Business Intelligence	Pentaho: 26.049	JasperReports: 3.292	Sofa statistics: 427
Desktop Publishing	ZintBarcodeGenerator: 1.473	TexMaths: 219	EqualX: 140
Report- Generators	Jaspersoft Studio: 37.725	BIRT Report Designer: 297	SuiteCRM: 240

³⁴ Eine genaue Definition und Differenzierung von Unternehmenssoftware existiert bis heute nicht und viele der genannten Begriffe werden synonym verwendet, vgl.: KEES, Alexandra. *Open Source Enterprise Software*. Springer Fachmedien Wiesbaden, 2015.

³⁵ Ebd.

³⁶ Alle hier verzeichneten Daten wurden das letzte Mal am 21.01.2016 aktualisiert. Vgl.: Sourceforge.net, 2016, Enterprise Software. [online]. 2016. [Abgerufen 21 Januar 2016]. Verfügbar unter: <http://sourceforge.net/directory/enterprise>

Knowledge-management	Freeplane: 4.081	OpenKM – DMS: 1.411	LogicalDOC- DMS: 1.188
Sales	Openbravo: 1.640	Chromis POS: 435	YetiForce CRM: 105

Tab. 2: Die beliebteste OSS von sourceforge.net im Bereich Unternehmenssoftware
Quelle: eigene Darstellung

Die bis zu sechsstelligen Downloadraten einzelner Programme zeigen deutlich dass es sich bei dem Komplex freier Software längst nicht mehr um ein Randphänomen handelt. Das Repository zählt insgesamt wöchentliche Downloadraten von bis zu 50 Millionen, wobei Applikationen wie MinGW (Minimal Gnu for Windows) 1,8 Millionen Downloads erreichen.

Die Innovationsfähigkeit von proprietärer Software wird mittlerweile als gering eingestuft, gerade auch in Bezug auf das Internet der Dinge, Cloud- Computing und Big-Data, also den wichtigsten Trendmärkten für die nächsten Jahre. Randal resümiert dass die Nutzung von OSS mittlerweile keine Alternative darstellt die man vernachlässigen kann, sondern eine Notwendigkeit wenn man sich auf dem neuen Informationsmarkt behaupten möchte.³⁷ Dieser Meinung ist auch Gus Robertson, CEO von NGINX, einer sehr erfolgreichen Webserver- Firma (Marktanteil Dezember 2015: 17%³⁸), dessen Produkt unter einer BSD- Lizenz vertrieben wird. Er spricht von einer vollständigen Diffusion beider Welten, dabei nennt er auch die Tendenz vieler proprietärer Firmen mittlerweile Open- Source- Projekte zu finanzieren, anzuleiten oder sogar eigene OSS zu entwickeln und zu vertreiben. Er verweist, dem Coverity- Scan ähnlich, auf die bessere Leistungsfähigkeit und Sicherheit von Open- Source Code.³⁹ Wie weitreichend seine Beobachtung ist lässt sich ahnen wenn man bedenkt dass sogar Microsoft, bisher vehementester Gegner von Open- Source (Steve Ballmer sprach 2001 vom „Linux- Krebsgeschwür“) ein linuxbasiertes, modulares Betriebssystem namens Azure- Cloud- Switch für Netzwerke in Rechenzentren angekündigt hat.⁴⁰ Dabei wird die hauptsächlich ökonomisch motivierte Aneignung von Open- Source vieler Firmen nicht weniger wichtig aber das Modell der Partizipation und der gemeinschaft-

³⁷ Randal 2015

³⁸ Vgl. News.netcraft.com, 2016, Web Server Survey | Netcraft. [online]. 2016. [Abgerufen 22 Januar 2016]. Verfügbar unter: <http://news.netcraft.com/archives/category/web-server-survey/>

³⁹ Laut seiner Aussage hat sogar das U.S. Department of Defense OSS als sicherer eingestuft, vgl. NGINX, 2015, The Evolution of Open Source: Interview with NGINX CEO Gus Robertson. [online]. 2015. [Abgerufen 22 January 2016]. Verfügbar unter: <https://www.nginx.com/blog/the-natural-evolution-of-open-source-an-interview-with-nginx-ceo-gus-robertson/>

⁴⁰ Vgl. Azure.microsoft.com, 2016, Microsoft showcases the Azure Cloud Switch (ACS) | Microsoft Azure Blog. [online]. 2016. [Abgerufen 26 Januar 2016]. Verfügbar unter: <https://azure.microsoft.com/de-de/blog/microsoft-showcases-the-azure-cloud-switch-acs/>

lichen Entwicklung für plattformunabhängige Anwendungen erhält zunehmend ein gewisses Prestige. Der wesentliche Konflikt beider Entwicklungsarten auf der sozialen Ebene bleibt zwar weiter bestehen, ist auf der pragmatischen Ebene in der IT-Community aber nicht mehr erkennbar.

2.2 Bibliotheksspezifische Open- Source- Software in den USA und Deutschland

Breeding unterteilt die vorhandenen Arten von Bibliothekssoftware in drei Klassen: evolutionär, revolutionär und alternativ.⁴¹ In die erste gehören die meisten proprietären aber auch freie Systeme mit dem Fokus auf vorhandenen Bestand und Verbesserung der Benutzeroberfläche mit Web 2.0- Anwendungen. Die zweite Klasse ist serviceorientiert auf den Ausbau der IT- Infrastruktur mithilfe moderner XML- Technologien konzentriert und die dritte zeichnet sich durch das Schaffen von APIs und Schnittstellen aus, mit dem Ziel der verbesserten Interoperabilität zur Internetinfrastruktur. Das größte Potential und die innovativsten und adaptionsfähigsten Entwickler- Gemeinschaften haben nach Breeding die alternativen, offeneren Systeme, womit Systeme auf Open- Source- Basis gemeint sind. Die Verwendung von OSS in Bibliotheken ist, wie in anderen Branchen auch, vielfältig und reicht von einzelnen Anwendungen auf dem Mitarbeiter- Desktop über einzelne Module, häufig Middleware, bis hin zu kompletten, cloudbasierten BMS/LIS, die aber eine Migration lokaler Systeme erforderlich machen. Die allgemeine Verschiebung von Services auf cloudbasierter Basis und der damit verbundene Abbau von lokaler Systemhardware sind globale Entwicklungen die sich auch im Bibliotheksbereich bemerkbar machen, wie Lubeek von OCLC bemerkt.⁴² Die USA hatte mit WorldShare von OCLC bereits 2010 das erste cloudbasierte System. Eine vollständige Übersicht über die Verwendung von OSS in Bibliotheken ist nicht möglich, da es in Deutschland keine aktuelle, nach standardisierten Facetten erhobenen Daten zum Thema gibt und die DBS und der BIX keine Angaben zu verwendeten Software- Produkten machen sondern lediglich digitale Dienstleistungen nennen. Marktübersichten zur Bibliothekssoftware betrachten üblicherweise keine OSS, oder nennen nur vereinzelte Beispiele.⁴³ Das letzte größere Forschungsvor-

⁴¹ Vgl.: Mittelbach, Jens, 2011, Zur Zukunft von Bibliothekssoftware. *Nbn-resolving.de* [online]. 2011. [Abgerufen 12 Januar 2016]. Verfügbar unter: <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-66156>

⁴² Vgl.: B-i-t-online.de, 2016, b.i.t.online: Bibliotheksfachzeitschrift mit Information und Technologie. [online]. 2016. [Abgerufen 12 Januar 2016]. Verfügbar unter: <http://www.b-i-t-online.de/daten/nuernberg/KN-2015-interview-lubeek>

⁴³ Eine vielzitierte Bachelorarbeit aus dem Jahr 2009 nennt lediglich ein einziges System: OpenBiblio, vgl.: REIMANN, Katlen. Aktuelle Marktübersicht des Bibliothekssoftwaremarktes in Deutschland unter Einbeziehung von Open Source Produkten. *Bachelorarbeit, Stuttgart. Online verfügbar*

haben zum Thema Open- Source- Software, durchgeführt von der TU Berlin, wurde 2008 beendet und betrachtete nicht den Aspekt der Bibliothekssoftware, sondern nur in Teilen angewandte Archivsoftware.⁴⁴ So sind die meisten der hier genannten Zahlen auf die Erhebungen des Herausgebers des Library Systems Report und der Library Technology Guides, Marshall Breeding, zurückzuführen. Wenn man sich die in den letzten 5 Jahren erschienenen Artikel zur Implementierung von OSS in Bibliotheken ansieht werden BMS/ILS- Anwendungen, gefolgt von Discovery- Systemen und Repositorien- Software genannt, was für die USA und Deutschland gleichermaßen gilt.⁴⁵ Typologisch sind die am häufigsten eingesetzten Systeme für Bibliotheken in die Felder Electronic Resource Management Systeme (ERMS), Digital Assets Management Systeme (DAMS), Portalsoftware (vor allem GUI- Recherche- und Discovery- Tools) und Link- Resolver einzuordnen. Besonders im Bereich kompletter BMS/ILS- Anwendungen ist OSS immer erfolgreicher. Zu nennen sind hier KOHA, Evergreen und potentiell auch das sich noch in der Evaluierungsphase befindliche Quali OLE. Auf letztgenanntes wird noch ausführlicher eingegangen. Für die USA nennt Singh die Verteilung dieser Systeme bezogen auf Bibliothekstyp und Bestandsgröße. So werden für wissenschaftliche, öffentliche und Spezialbibliotheken mit weniger als 100.000 Medieneinheiten KOHA, KOHA- Bywater, Evergreen und Evergreen- Equinox benutzt, bei einer Bestandsgröße bis zu 1.000.000 ME fast ausschließlich Evergreen und bei einer Größe von mehr als einer Million ME auch Quali OLE.⁴⁶ KOHA- Bywater ist dabei ein Beispiel für eine privatwirtschaftliche Firma, die professionelle Hilfe bei der Implementierung und Entwicklung des ILS als Service anbietet. Die Auslagerung der eigentlichen Softwareentwicklung und des Support ist ein relativ häufiges Modell bei der Nutzung von OSS. Zumindest für KOHA lässt sich in Deutschland eine allmähliche Verbreitung in wissenschaftlichen Bibliotheken nachvollziehen, aber viele Implementierungsphasen sind noch nicht abgeschlossen und entsprechende Daten sind noch nicht verfügbar. Der jährliche Library Automation Report von Marshall Breeding sammelt Daten über den Automatisierungsprozess von Bibliotheken und listet die jeweils dazu verwendete Software auf. Wirklich aussagekräftig sind diese Zahlen für

unter: <http://opus.bszbw.de/hdms/volltexte/2008/652/pdf/Bachelorarbeit.pdf> (letzter Zugriff: 21.01. 2016), 2008.

⁴⁴ Die gesammelten Beiträge sind im Volltext online verfügbar: Informatik und Gesellschaft, TU Berlin, 2016, Open Source Jahrbuch. *Opensourcejahrbuch.de* [online]. 2016. [Abgerufen 13 Januar 2016]. Verfügbar unter: <http://www.opensourcejahrbuch.de/project/index.html>

⁴⁵ Für Deutschland siehe: FISCHER, Katrin; MEIßNER, Holger. Open Source All Inclusive-Betriebs- und Entwicklungsmodelle für das Bibliothekssystem Koha. Für Amerika siehe: SINGH, Vandana. Why migrate to an open source ILS? Librarians with adoption experience share their reasons and experiences. *Libri*, 2013, 63. Jg., Nr. 3, S. 206-219.

⁴⁶ Singh 2013

Deutschland jedoch nicht, da die Anzahl mit 495 teilnehmenden Bibliotheken zu gering ist. Im Gegensatz zu Amerika wird KOHA in Deutschland fast ausschließlich von wissenschaftlichen Bibliotheken eingesetzt.⁴⁷ Breeding schätzt den gesamten Marktanteil von Open- Source- Software für öffentliche Bibliotheken in Amerika (wiederholt Koha und Evergreen) auf ca. 12%, für die wissenschaftlichen Bibliotheken gibt es keine genaue Angaben.⁴⁸ Er geht nicht davon aus das für akademische Einrichtungen OSS eine tragende Rolle in den nächsten Jahren in Amerika spielen wird, bemerkt aber dass mit Evergreen eine vielversprechende Alternative auf dem Markt ist die sich bei mehr erfolgreichen Implementierungen mit einem spürbaren Marktanteil durchsetzen könnte. Mehrere Open- Source- Module sind aber teilweise für das Funktionieren proprietärer BMS/ILS- Systeme notwendig. Beispiele hierfür sind, das YAZ Z39.50 toolkit der Firma Index Data, der Apache Web server, Apache Tomcat, Apache Axis, Linux und die Repositorien Software Fedora und Dspace sowie das Discovery- Interface VuFind. Die Erweiterungen (Plugins) Scriblio und Blacklight sind keine eigenständigen Systeme sondern gehören zu den verwendeten CMS Wordpress und Drupal. Immer wieder genannt wird KOHA, welches 2014 von 94 Bibliotheken neu implementiert wurde⁴⁹ und weiter von wissenschaftlichen Bibliotheken mit einer Bestandsgröße von weniger als 100.000 ME immer häufiger eingesetzt wird. Die folgende Tabelle mit Daten des Portals Librarytechnology.org stellt nur eine Annäherung dar und ist weder repräsentativ noch vollständig.

Benutzte OSS im Jahr 2016	USA (143.264 Bibl. gesamt)	DEUTSCHLAND (495 Bibl. gesamt)
ABCD (DAMS)	k.A.	1
Allegro-C (DBMS)	k.A.	54
Archivists Tool Kit (DAMS)	62	1
Blacklight (Discovery- Interface/Drupal- Plugin)	19	k.A.
DSpace (Repositorien)	1651	25
EPrints (Repositorien)	k.A.	k.A.
Evergreen (BMS/ILS)	1337	k.A.
Fedora (Repositorien)	143	9
Greenstone (Digitale Sammlungen)	5	k.A.
KOHA (BMS/ILS)	1636	28

⁴⁷ Nach einer Registrierung lässt sich die erweiterte Suche nutzen. Vgl.: Breeding, Marshall, 2016, Library Technology Guides: Documents, Databases, News, and Commentary. *Librarytechnology.org* [online]. 2016. [Abgerufen 13 Januar 2016]. Verfügbar unter: <http://librarytechnology.org>

⁴⁸ Vgl.: American Libraries Magazine, 2015, Library Systems Report | American Libraries Magazine. [online]. 2015. [Abgerufen 13 Januar 2016]. Verfügbar unter: <http://americanlibrariesmagazine.org/2015/05/01/library-systems-report/>

⁴⁹ Ebd.

Kuali OLE (BMS/ILS)	4	k.A.
Omeka (Digitale Sammlungen)	7	k.A.
OPALS (BMS/ILS)	1445	2
OpenBiblio (BMS/ILS)	6	11
Scriblio (Discovery- Service/ WP- Plugin)	2	k.A.
SOPAC (Discovery- Service)	13	k.A.
VuFind (Discovery- Interface)	277	4

k.A. = keine Angabe

Tab. 3: Verwendete OSS in Bibliotheken in USA/Deutschland 2016

Quelle: eigene Darstellung in Anlehnung an Breeding 2016

Legt man die von Breeding gesammelten Daten zugrunde benutzen 4,6% der amerikanischen und 27.3% der deutschen Bibliotheken mindestens ein OSS- Tool. Laut einer Befragung aus dem Jahr 2014, in der 77 Forschungsbibliotheken der Association of Research Libraries (ARL) nach dem Stand des Einsatzes freier Software befragt wurden⁵⁰, gaben 97% an mindestens ein OSS- Tool implementiert zu haben. Davon benutzten 8.1% ein ILS, 53.11% ein DAM, 75.4% entsprechende Software für das institutionelle Repositorium und 26.0% einen quelloffenen Discovery- Service⁵¹. Nach den Listen von Winterhalter und Lohnert basieren 14% der 2014 verwendeten Softwareapplikationen in deutschen Bibliotheken auf OSS,⁵² wobei den größten Teil Literaturverwaltungsprogramme wie Zotero einnehmen und lediglich ein Dokumentenserver und zwei BMS/ILS genannt werden. Folgend werden einige OSS- Anwendungen im Bereich der BMS/ILS, der Repositorien und der Discovery- Services näher beschrieben die sowohl in Amerika als auch in Deutschland eingesetzt werden.

Auf dem Markt der BMS/ILS hat sich KOHA als eine der meistbenutzten OSS etabliert.⁵³ Das System wird seit 1999 entwickelt und hat seinen Ursprung in Neuseeland. In einer kleinen Stadtbibliothek sollte ein proprietäres System ersetzt werden um die Jahresumstellung auf das Jahr 2000 ohne größere Konflikte zu gewährleisten. KOHA wurde 2000 unter der GNU General Public License (GPL) veröffentlicht und wird heute von einer großen internationalen Community mit verschiedensten Kollabo-

⁵⁰ Vgl. THACKER, Curtis; KNUTSON, Charles. Barriers to Initiation of Open Source Software Projects in Libraries. *Code4Lib Journal*, 2015, Nr. 29.

⁵¹ Für eine volle Übersicht der OSS- Nutzung durch ARL- Bibliotheken siehe Anhang II, Abb. 4, S. IX.

⁵² Auch sie verweisen auf die Unvollständigkeit und geben keine Garantie auf Richtigkeit der Angaben, vgl.: WINTERHALTER, Christian. Die EDV und ihre Systeme: Software. In: ALBERS, Christoph. *Erfolgreiches Management von Bibliotheken und Informationseinrichtungen: Fachratgeber für Bibliotheksleiter und Bibliothekare*. Verlag Dashöfer, 2013.

⁵³ KOHA wird laut Breeding weltweit in 3258 Bibliotheken eingesetzt, vgl.: Breeding, Marshall, 2016, Library Technology Guides: Documents, Databases, News, and Commentary. *Librarytechnology.org* [online]. 2016. [Abgerufen 16 Januar 2016]. Verfügbar unter: <http://librarytechnology.org>

rationen getragen.⁵⁴ Es ist plattformunabhängig und damit lauffähig auf Linux, Mac oder Windows. Es kann in Bibliotheksverbünde, Bibliotheken mit Zweigstellen sowie über ein Extra- Modul von einer Spezialbibliothek integriert werden. Wie die meisten BMS/ILS deckt KOHA die Module OPAC, Zeitschriften, Erwerbung, Ausleihe und Katalogisierung ab, wobei sowohl OPAC als auch das Backend- System über den Browser konfiguriert werden. Die verwendeten Komponenten basieren auf Linux, Apache, MySQL sowie den Programmiersprachen Perl und Zebra. Die verfügbaren Schnittstellen sind Z39.50, SIP2, LDAP, und ILS-DI. OAI- PMH betreffend erlaubt KOHA aufgrund des nativen MARC- Frameworks nur die Ausführung als Datenlieferant (Data Provider) und nicht als Dienstanbieter (Service Provider/Harvester). MARC21, UNIMARC und NORMARC sind integriert. In Deutschland wird KOHA vom BSZ Baden- Württemberg als Software- as- a- Service (SaaS) angeboten, welche Hosting und Support übernehmen. Besonders wichtig ist hier die Unterstützung bei der Datenmigration, einer der wichtigsten Services bei der Umstellung auf eine neue IT-Infrastruktur und keine Selbstverständlichkeit.⁵⁵ Die TFH Wildau gehörte zu den ersten Anwendern des Systems in Deutschland und installierte dieses im Dezember 2012.

Ein weiteres web- und cloudbasiertes System derselben Sparte das auch für größere Bibliotheken (mehr als 1 Millionen ME) in Amerika und Deutschland interessant ist⁵⁶, ist das unter der OSS- Lizenz ECL2 veröffentlichte Quali OLE.⁵⁷ Das Kürzel OLE steht für Open Library Environment und definiert damit genau worum es hier geht. Als momentan größte Forschungsbibliothek in den USA die diese OSS einsetzt gilt die Universitätsbibliothek Chicago die eine Bestandsgröße von 11.397.412 ME (Stand 2016) hat. Als typisches Setup benutzt die Bibliothek den EBSCO Discovery- Service mit dem OSS- Interface VuFind als Discovery- Service (in Deutschland ist auch der Anschluss des an der hbz entwickelten DigiBib- Moduls möglich). Die Basis bildet

⁵⁴ Vgl. Koha-community.org, 2016, Official Website of Koha Library Software. [online]. 2016. [Abgerufen 26 Januar 2016]. Verfügbar unter: <https://koha-community.org/>

⁵⁵ Vgl. Bsz-bw.de, 2016, Bibliothekssysteme im Bibliotheksservice-Zentrum Baden-Württemberg: Koha. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <https://www.bsz-bw.de/bibliothekssysteme/koha.html>

⁵⁶ Der mögliche Einsatz von Quali OLE mit Pilotprojekten in Deutschland wird durch das hbz und GBV evaluiert, vgl.: Hbz-nrw.de, 2016, hbz — Entwicklungspartnerschaft zwischen hbz, VZG und der Quali OLE Community. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: https://www.hbz-nrw.de/aktuelles/entwicklungspartner_ole

⁵⁷ Für die folgenden Zahlen und Beschreibungen des Systems siehe: Hbz-nrw.de, 2016, hbz — Das Bibliotheksmanagementsystem Quali OLE – Überblick und aktueller Stand. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: https://www.hbz-nrw.de/dokumentencenter/veroeffentlichungen/bms_kuali_ole_2015.pdf/view

die Quali- Enterprise- Architektur.⁵⁸ Geschrieben ist Quali in der Programmiersprache Java 7 und hat den für ein System dieser Komplexität üblichen Umfang von ca. 1 Millionen Quellcode- Zeilen. Die Funktionen von Quali bauen auf vier Modulen auf: Medienauswahl und Erwerbung (beinhaltet auch Funktionen zur Lizenzierung und Verwaltung von E- Ressourcen, der Haushaltsverwaltung und der Rechnungsbearbeitung), Metadatenmanagement (Verwaltung von Lokal- und Exemplardaten), Ausleihe (Benutzerdaten, Mahnungen, Gebühren, Nutzerfunktionen Vormerken/ Bestellen) und Systemintegration (Rechte- und Identitätsmanagement).⁵⁹ Workflows für E- und Printmedien sind enthalten. Der Anschluss an die Global Open Knowledge Database (GOKb) soll als Standardfeature noch dazu kommen, was den Zugang zu dessen Metadaten, vornehmlich E- Journals und E- Books, über einen bidirektionalen automatisierten Datenimport sichern wird. Eine der Neuerungen ist die flexible Modulstruktur unabhängig von bestimmten Medientypen. Dies erlaubt einen höheren Grad an Personalisierung und Skalierbarkeit innerhalb unterschiedlicher Geschäftsgänge. Vorhandene Schnittstellen sind neben der schon genannten bidirektionalen GOKb- Anbindung und SIP2, SMTP, RESTful und EDIFACT (automatische Bestellungen und Rechnungen), NCIP (automatische Bestelldatensätze) und Z39.50 sowie SRU/SRW für den Datenimport und die Datenbankabfrage. Unterstützte Datenformate sind MARCXML, MARC21, Dublin Core und das eigene OLEML- Format für Lokal- und Exemplardaten. Noch als Standard hinzukommen sollen EAD, MODS, METS und BIBFRAME für die semantische Transformation der Daten. Noch nicht ausgereift ist das Reporting Framework für Dokumentation und Statistik. Nach einem neuen Ansatz sollen aber über ODBC- Schnittstellen systemextern arbeitende Statistik- Programme eingebunden werden können. Wie auch KOHA ist das System ohne Vendor- Lock- In plattformunabhängig einsetzbar (Windows, MacOS X, Linux). Die eigentliche Programmierung erfolgt durch die privatwirtschaftlichen Software- Firma HTC global services, welche laut eigenen Angaben etwa 2000 Kunden weltweit betreut und als Host für alle Mitglieder der Quali- Foundation fungiert.⁶⁰ Schweitzer und Kemner- Heek kamen auf dem 104. Bibliothekartag zu dem Schluss dass Quali OLE für den Einsatz in Deutsch-

⁵⁸ Vgl. Quali Foundation, info @ kuali.org, 2016, Quali Enterprise Workflow (KEW). *Kuali.org* [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <https://www.kuali.org/rice/modules/kew>

⁵⁹ Für eine detailliertere Übersicht der Funktionen siehe auch: Quali Foundation, info @ kuali.org, 2016, Functionality by Module. *Kuali.org* [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <http://www.kuali.org/ole/modules>

⁶⁰ Vgl. Htcinc.com, 2016, HTC Global Services. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <http://www.htcinc.com/home.asp>

land durchaus geeignet ist.⁶¹ Beide Systeme zeichnen sich durch einen hohen Grad an Internationalisierung aus, sind größtenteils web- und cloudbasiert und besitzen eine ausreichende Skalierbarkeit und Schnittstellen für die Einbindung externer Ressourcen wie Discovery- oder Statistik- Programme. Im Fall von Quali OLE liegt darüber hinaus die komplette Entwicklung in den Händen von zwölf Partnerbibliotheken und Verbänden mit einem selbst gewählten Dachverband. Dies beinhaltet genaue über verschiedene Gremien abgeklärte Programmiervorgaben, was bedeutet dass lokale Anforderungen frühzeitig in den Implementierungsprozess einfließen können. Ein nicht zu unterschätzender Vorteil wenn man mühseliges Reverse- Engineering vermeiden will, zu dem einige Bibliotheken bei proprietären Systemen durchaus gezwungen werden können, welche die eigentlich gewünschte Funktionalität oft nicht zulassen.

Die wichtigsten Charakteristika die moderne institutionelle Repositorien aufweisen müssen sind die Interoperabilität zu web- basierten Systemen, eine hohe Flexibilität und Anwendungsbreite hinsichtlich der Datentypen und eine hohe Modifizierbarkeit, da die lokalen Veröffentlichungspraktiken auf die jeweilige Institution und deren veröffentlichenden Mitglieder zugeschnitten sind die aus allen Fachrichtungen stammen können die ein breites Spektrum an Primär- und Sekundärdatenmodellen aufweisen. Weiterhin sind eine OAI-PMH- Schnittstelle und Funktionen wichtig die eine Veröffentlichung der Datensätze als Linked Open Data unterstützen. Die Einsichtbarkeit des Codes und die damit zusammenhängende Dokumentation der Änderungen von durchgeführten Prozessen sind für Archive von Vorteil wenn es darum geht Migrationsaktivitäten nachzuvollziehen und diese Prozesse letztlich auch zu archivieren.⁶² Deshalb sollten OSS- Module gerade in Hinsicht auf die Langzeitarchivierung in Betracht gezogen werden. Die folgend beschriebenen Systeme Fedora und DSpace sind in den USA und Deutschland häufig eingesetzte Systeme. DSpace ist dabei mit etwa 3753 registrierten Repositorien⁶³ die meistgenutzte OSS- Software in diesem Sektor weltweit und vielen Nutzern auch aus dem kommerziellen Bereich von Informationsdienstleistungen. Vergleicht man die rein für Open- Access- Repositorien benutzte Software

⁶¹ Vgl. Schweitzer, Roswitha and Kemnek- Heek, Kirstin, 2016, Das Open Source Bibliothekssystem Quali OLE : Ergebnisse der gemeinsamen Evaluation durch GBV und hbz. In : *104. Deutscher Bibliothekartag in Nürnberg* [online]. Nürnberg : Berufsverband Information Bibliothek. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <https://opus4.kobv.de/opus4-bib-info/frontdoor/index/index/docId/2078>

⁶² Vgl. FUNK, Stefan E.; NEUBAUER, Mathias. Formatcharakterisierung. In: NEUROTH, Heike. *nestor Handbuch. Boizenburg: Verlag Werner Hülsbusch, 2009.*

⁶³ Aktueller Stand 2016, vgl.: [Registry.duraspace.org](http://registry.duraspace.org), 2016, DSpace User Registry | DuraSpace. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <http://registry.duraspace.org/registry/dspace/>

über das OpenDOAR- Portal nimmt DSpace auch dort mit 1307 registrierten Repositorien einen größeren Abdeckungsbereich von 43,4% ein, gefolgt von EPrints mit 411 Repositorien und damit 13,6%. Das noch folgend näher beschriebene Repository Fedora⁶⁴ ist mit 42 Einträgen und damit 1,4% am unteren Ende der Skala in diesem Verzeichnis finden.⁶⁵ Weltweit sind laut Duraspace etwa 333 Fedora- Repositorien registriert.⁶⁶ DSpace wird aktuell von Duraspace als not- for- Profit- Organisation verwaltet und entwickelt war aber ursprünglich 2002 von den Hewlett- Packard Labs in Zusammenarbeit mit dem MIT veröffentlicht worden. Weitere Mitglieder der heute verantwortlichen DSpace- Committers- Group umfassen außerdem Personen von größeren Organisationen wie OCLC, der TU Berlin oder der University of Edinburgh. Mitglieder dieser Gruppe sind es auch die direkt auf den Quellcode Zugriff haben und diesen ändern dürfen, Vorschläge und Änderungen der Community einarbeiten, Fehler (Bugs) und Sicherheitsrisiken auflösen, die Kommunikation mit der Gemeinschaft und den Endnutzern über IRC und Mailinglisten in Bewegung halten und die in der Roadmap festgelegten Releases veröffentlichen.⁶⁷ Committer wird nur, wer kontinuierlich über einen längeren Zeitraum zu dem Projekt beigetragen hat in Form von Fehlerberichten (Bug reports⁶⁸), Teilnahme an Entwicklertreffen und Mitarbeit am Code in Form von pull requests. Dies entspricht den unausgesprochenen Gepflogenheiten in etablierten Open- Source- Communities (auf weitere soziale Normen im Kontext mit Software- Entwicklern wird in einem späteren Kapitel noch eingegangen). Typologisch gehört DSpace wie die meiste Software in diesem Bereich zu den Digital- Asset- Management- Systemen (DAM). Die drei Kernkomponenten sind die Suchmaschine Lycene von Apache, das CRNI- Handle- System (verwendetets Protokoll: TCP) zur Verwaltung von DOI's und die für alle Repositorien unumgängliche Schnittstelle OAI- PMH. Die zugrunde liegende Programmiersprachen sind Java und JSP., das unterstützte Metadaten- Format ist Dublin- Core.

⁶⁴ Es existiert eine Linux- Desktop- Distribution mit gleichem Namen, die von der Firma Red- Hat- Linux entwickelt und vertrieben wird und mit dem Repository nur den Namen gemein hat.

⁶⁵ Vgl. MILLINGTON, Peter, 2016, OpenDOAR - Home Page - Directory of Open Access Repositories. *OpenDoar.org* [online]. 2016. [Abgerufen 18 Januar 2016]. Verfügbar unter: <http://www.opendoar.org/>

⁶⁶ Stand Januar 2016, vgl.: Registry.duraspace.org, 2016, Fedora User Registry | DuraSpace. [online]. 2016. [Abgerufen 18 January 2016]. Verfügbar unter: <http://registry.duraspace.org/registry/fedora>

⁶⁷ Vgl.: Wiki.duraspace.org, 2016, Dashboard - DuraSpace Wiki. [online]. 2016. [Abgerufen 18 Januar 2016]. Verfügbar unter: <https://wiki.duraspace.org/dashboard.action>

⁶⁸ Um die Software an sich zu verbessern und die Community zu unterstützen gibt es viele freiwillig unterhaltene, frei zugängliche Bug- Datenbanken (Bug Tracking Systems).

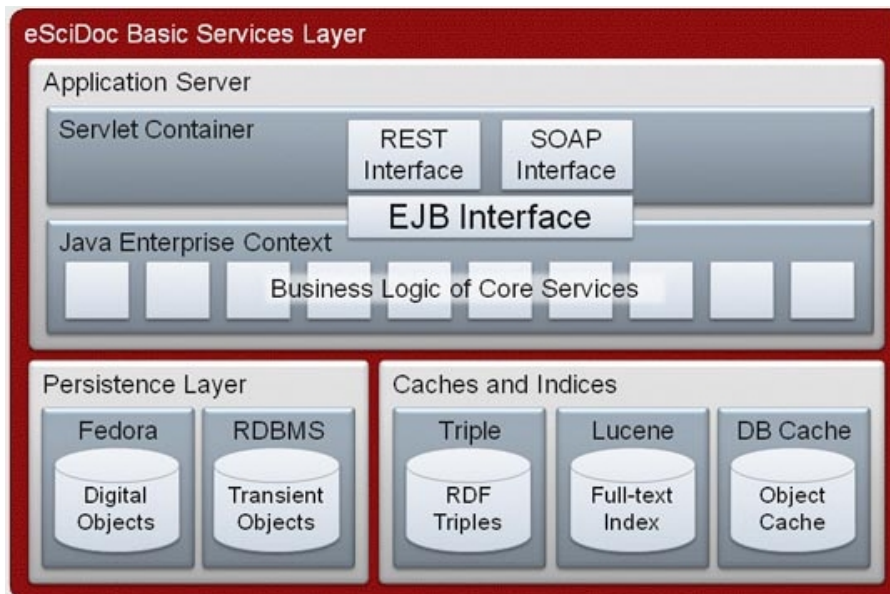


Abbildung 1: Einbindung von Fedora in die Persistenzschicht der eSciDoc-Infrastruktur

Quelle: <https://www.esdoc.org/JSPWiki/en/InfrastructureArchitecture>

Das ebenfalls in Java programmierte Fedora (Flexible Extensible Digital Object Repository) ist als Kernkomponente und Infrastruktur und weniger als Endnutzer- Tool zu verstehen, ganz im Sinne der Definition moderner Repositorien: „[...] the repository itself is just a background service, providing storage, persistent identification, preservation, and discovery of the content.“⁶⁹ Für die Forschungs- und Publikationsplattform eSciDoc die sich auf e- Research spezialisiert hat, gelten dabei Anforderungen die über ein reines Repository hinaus gehen. Das Datenmodell muss die hohe Diversität von Primärdaten in Betracht ziehen und arbiträre Metadaten- Profile (lokal definierte Attribute) unterstützen. Jeder Forscher der mit der mit der Plattform arbeitet und publiziert kann die Funktionalitäten in den Bereichen Recherche, Speicherung, Authentifizierung und die Zugangsrechte selbst manipulieren. Die meisten Programmier- und Skriptsprachen werden über einfach zu bedienende API's unterstützt. Virtuelle Arbeitsgruppen können über das integrierte Shibboleth- Authentifizierungsprogramm die Funktionen der Plattform nutzen. DSpace ist im Gegensatz zu dem kaum vorkonfigurierten Fedora ein Komplettsystem, dass „out of the box“ funktioniert. Beide sind OAIS- konform, unterstützen Linked- Data- Technologien wie die Transformation der Datensätze in RDF- Tripel und werden in den USA und Deutschland als Publikations- und Dokumentenserver für Universitätsbibliotheken verwendet. Im Fall von DSpace lässt sich aufgrund der Verbreitung über alle Bibliotheks- und Archivtypen hinweg von

⁶⁹ Für dieses Zitat und die folgenden Ausführungen vgl.: RAZUM, Matthias, et al. eSciDoc infrastructure: a fedora-based e-Research framework. In: *Research and Advanced Technology for Digital Libraries*. Springer Berlin Heidelberg, 2009. S. 227-238.

einer vollständigen Implementierung in die Informationsinfrastruktur Amerikas sprechen. In Deutschland ist mit ca. 100 Installationen⁷⁰ OPUS die verbreitetste OSS für institutionelle Repositorien und umfasst Nutzungen als Hochschulschriftenserver mit und ohne Bibliographiefunktion, reiner Bibliographieserver sowie internen Dokumentenserver. Bei einer Gesamtmenge von derzeit registrierten 177⁷¹ institutionellen Repositorien in Deutschland bedeutet dies einen Anteil von 56,5%. Der KOBV, BSZ und das hbz sind als Host für die Entwicklung selbst verantwortlich. Mehr als die Hälfte der gesamten Opus- Instanzen wird von Verbänden gehostet. Wie auch die anderen beschriebenen Beispiele trat OPUS mit der zunehmenden Etablierung des Open- Access-Gedankens Ende der 1990er Jahre in Erscheinung und baut auf den OSS- Komponenten Apache, MySQL und PHP5 auf. Zur interoperablen Struktur gehören die Einbindung der DDC- Sachgruppen und die GND. OPUS wurde mithilfe der OSS- Eclipse-IDE geschrieben, einer in der Linux- Welt sehr verbreiteten Entwicklungsumgebung.

Laut Breeding wird das Discovery- Interface VuFind in Deutschland von der HTWK Leipzig, UB Leipzig, der Bibliothek der TUHH Hamburg und der UB Freiburg eingesetzt.⁷² Die Homepage verzeichnet insgesamt ca. 20 Installationen.⁷³ Fast alle bauen auf der Architektur eines proprietären Discovery- Services wie Ebsco- Discovery auf. Die breite Inkludierung von externen Webressourcen durch VuFind umfasst den Online- Buchhandel genauso wie kommerzielle Suchmaschinen. Das Such- Fenster ist mit einfacher Suchbox und einem einzigen Drop- Down Menü simpel und übersichtlich gestaltet. Es lassen sich folgende Facetten bei der einfachen Suche anwenden: Titel, Zeitschriftentitel, Verfasser, Schlagwort, Signatur, und ISBN/ISSN. Formateinschränkungen oder Boolesche Operatoren lassen sich über die erweiterte Suche anwenden. Typische Web 2.0 Tag- und Kommentierungsfunktionen, Biographieinformationen zum Autor, Google- Book- Previews oder Favoritenlisten sind (wahlweise) integriert und der übliche Standard in Open- Source- und proprietären Discovery- Interfaces. Das neben VuFind zumindest in den USA öfter verwendete Blacklight hat den gleichen Funktionsumfang und beruht genauso auf der Programmiersprache Ruby on Rails statt des in OSS- Anwendungen üblichen PHP- Codes. Das hohe Adoptionsver-

⁷⁰ Vgl.: CONRAD- REMPEL, Steffi, 2015, Opus 4 in Zahlen und Fakten beim KOBV. In : 104. *Deutscher Bibliothekartag* [online]. Nürnberg. 2015. [Abgerufen 19 Januar 2016]. Verfügbar unter: <https://opus4.kobv.de/opus4-bib-info/frontdoor/index/index/docId/1936>

⁷¹ Millington 2016

⁷² Breeding 2016

⁷³ Vgl.: [Vufind.org](https://vufind.org), 2016, community:installations [VuFind Documentation]. [online]. 2016. [Abgerufen 19 Januar 2016]. Verfügbar unter: <https://vufind.org/wiki/community:installations>

mögen und die Eigenart als einziges Interface für die unterschiedlichsten parallel laufenden Services aus dem proprietären und freien Softwaresektor fungieren zu können, werden von vielen Fachleuten in Amerika als das wichtigste Merkmal bei der Entscheidung für VuFind gesehen.⁷⁴

Wie das Beispiel der TUHH andeutet (Abb. 2), ist für die meisten Bibliotheken ein Mix aus Anwendungen beider Softwarearten für verschiedene Aufgaben die Regel. Auffallend ist dass gerade wenn es um die Anbindung an die moderne Internetinfrastruktur geht OSS als Bestandteil von Middleware gewählt wird. Die meisten BMS-Komplettsysteme oder Kernanwendungen für das Datenmanagement bleiben aber, wie verschiedene Stichproben gezeigt haben in beiden Ländern proprietär. Für die hier gezeigte mittelgroße Universitätsbibliothek sind die OSS WordPress, DSpace und VuFind Bestandteil der direkt mit dem Netz kommunizierenden Schicht in einem ansonsten von proprietärer Software dominierten technologischen Profil.

Technology Profile		
	Product Name	Year Contracted
Current Automation System	LBS Lokaal Bibliotheek System	2000
Previous Automation System	SISIS	1994
Discovery Service (w/Index)	Primo	2014
Discovery Interface	VUfind	2010
OpenURL Link resolver	SFX	2005
Institutional Repository	DSpace	
Web Content Management	WordPress	
Item ID Type	RFID	
RFID Provider	Bibliotheca	
RFID Tag Type	HF	
Self-Check	Bibliotheca	
This library is responsible for the procurement of the library automation system.		

Abbildung 2: Das technische Profil der TUHH

Quelle: Breeding 2016

Das vielleicht wichtigste und nötigste Charakteristikum neuerer Bibliothekssoftware seit 2010 ist die zunehmende Interoperabilität mit web- affinen Workflows und damit die Auseinandersetzung mit vielschichtigen Formaten und Web- Vokabularen, wobei das größte Augenmerk auf FRBR, RDA, XML und RDF liegen dürfte. Für Metadaten

⁷⁴ Vgl.: SEAMAN, Graham. Adapting VuFind as a Front-end to a Commercial Discovery System. *Ariadne*, 2012, Nr. 68.

sind Dublin Core, MARC21, TEI, EAD oder CIMI wichtig, für Bilder und Grafiken TIFF, GIF und JPEG- Formate. Gerade die immer noch andauernde Verhaftung an MARC erschwert dabei vielen amerikanischen Bibliotheken und der angewandten Software den Schritt in die gleiche Ebene die Google bereits 2008 mit dem Knowledge graph erfolgreich betreten hat, eine Tendenz die gefährlich sein kann, wie Ayre betont: „[...] we are isolated technologically because our attachment to MARC makes it impossible to participate in a meaningful way with the rest of the interconnected, web-based world.“⁷⁵ Obwohl es keine genauen Zahlen oder eine repräsentative Übersicht gibt, geht man beim Einsatz von OSS im Gegensatz zu proprietärer Software in größeren Bibliotheksverbänden von Einsparungsvolumen in Millionenhöhe aus.⁷⁶

3 Problematik I - Urheberrecht, Patentrecht und Softwarelizenzierung

Die rechtlichen Rahmenbedingungen zum Einsatz von freier Software sind vielschichtig und einer der wichtigsten Aspekte die es bei einer Implementierung zu beachten gibt. Da die meiste OSS innerhalb oder als Ergänzung von Closed- Source- Software in Erscheinung tritt ist die genaue Kenntnis der Regelungen zum geistigen Eigentum eine der Grundvoraussetzungen für einen reibungslosen Einsatz in Institutionen und Betrieben, da seit dem Ende der 1970er Jahre in Amerika und Deutschland Computerprogramme unter diesem Abschnitt der Rechtssysteme mit länderspezifischer Gewichtung behandelt werden. Die das geistige Eigentum betreffenden Rechtssparten im Kontext von Software sind das Urheber-, Patent- und Markenrecht, aber auch das Haftungs- und Wettbewerbsrecht sowie im Zusammenhang mit Lizenzen das Vertragsrecht.

3.1 Urheberrecht

„We can build, or architect, or code cyberspace to protect values that we believe are fundamental. Or we can build, or architect, or code cyberspace to allow those values to disappear. There is no middle ground.“⁷⁷

⁷⁵ Vgl: AYRE, Lori Bowen. MARC isn't Dying Fast Enough. *Collaborative Librarianship*, 2015, 6. Jg., Nr. 4, S. 175-177.

⁷⁶ So nennt Wheeler für den Bibliotheksverbund der Georgia Public Library, die auf Evergreen umgestiegen ist eine jährliche Kostenersparnis von ca. 3 Mill. \$. Die Abhandlung untersucht die Kosten von OSS unter Einbeziehung der Skalierbarkeit, TCO und weiteren Faktoren. Für eine 2015 aktualisierte Fassung der 2005 erschienen Abhandlung vgl.: Dwheeler.com, 2016, Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!. [online]. 2016. [Abgerufen 28 January 2016]. Verfügbar unter: http://www.dwheeler.com/oss_fs_why.html

⁷⁷ Vgl. LESSIG, Lawrence. Code: And Other Laws of Cyberspace, Version 2.0. 2006.

Wie Boyle anmerkt haben diejenigen in der Informationsgesellschaft die Macht, die die Rahmenbedingungen des geistigen Eigentums festlegen und diejenigen die verstehen daraus Nutzen zu ziehen. Er legt dar dass es das geistige Eigentum ist, dass festlegt wie moderne Informationsgesellschaften das (bildungs-) politische, kulturelle und wissenschaftliche Potential moderner Technik und des Internet letztendlich nutzen und nicht die Auseinandersetzung über die Regulation von „schmutzigen“ Webinhalten.⁷⁸ Legt man diese Folgerungen Boyles und die Feststellung Lessigs zugrunde dass es die Beschaffenheit des Codes oder Internetinfrastruktur ist, die bestimmt wer auf welche Dokumente wann Zugriff hat, wird klar wie eng die Geschichte der Softwarelizenzierung mit der des geistigen Eigentums verwoben sein kann.

Wie bereits in einem vorigen Kapitel angedeutet entstand eigentlich proprietäre Software erst 1978. Grundlage dessen war die Empfehlung der Commission on New Technology Uses of Copyrighted Works⁷⁹ (CONTU) Computerprogramme wie literarische Werke zu behandeln und damit dem Schutz des Copyright zu unterstellen. Das Revidierte Berner Übereinkommen übernahm diese Tendenz 1979. Aber erst nach der von der Europäischen Kommission veröffentlichten Richtlinie zum Schutz von Computerprogrammen 1991 fand mit einem Eintrag in das Urheberrechtsgesetz schließlich 1993 der Schutz von Programm- Code in Deutschland eine rechtswirksame Basis unter Abschnitt 2 „Das Werk“, genauer der § 2 Abs 1. UrhG „Geschützte Werke“. Hier heißt es im Wortlaut: Zu den geschützten Werken der Literatur, Wissenschaft und Kunst gehören insbesondere: Sprachwerke, wie Schriftwerke, Reden und Computerprogramme.⁸⁰ Zusammen mit dem Abschnitt 8 „Besondere Bestimmungen für Computerprogramme“ § 69a – 69g UrhG bildet dies das Rückgrat der der deutschen Rechtsauffassung von Software. Im § 4 UrhG Abs.2 findet man schließlich die Trennung von Software und dem durch sie generierten Inhalt und Design auf Datenbanken bezogen.⁸¹ Eine der wichtigsten Unterscheidungen im deutschen und amerikanischen Copyright ist die Tatsache das in den USA ein Abtreten der Rechte am geistigen Eigentum durch den Autor pauschal möglich ist, in Deutschland verbleibt dieses immer beim Urheber.

⁷⁸ Vgl. BOYLE, James. A Politics of Intellectual Property: Environmentalism for the Net?. *Duke Law Journal*, 1997, 47. Jg., Nr. 1, S. 87-116.

⁷⁹ Der Abschlussbericht der Kommission ist online frei verfügbar, vgl.: Digital-law-online.info, 2016, CONTU Final Report. [online]. 2016. [Abgerufen 20 Januar 2016]. Verfügbar unter: <http://digital-law-online.info/CONTU/PDF/index.html>

⁸⁰ Vgl. Gesetze-im-internet.de, 2016, UrhG - Einzelnorm. [online]. 2016. [Accessed 20 January 2016]. Available from: https://www.gesetze-im-internet.de/urhg/_2.html

⁸¹ Ebd.

Obwohl internationale Verträge wie der auch von der EU 2001 ratifizierte Urheberrechtsvertrag der World Intellectual Property Organisation (WIPO) von 1996 als sehr einflussreich gelten, weist Kuhlen darauf hin, dass es nicht UN-Organisationen wie diese sind welche die Regeln des Intellectual Property Rights kontrollieren. Er sieht den wirklichen Akteur in der World Trading Organisation und durch sie forcierten Abkommen GATS (General Agreement on Trade in Services) und TRIPS (Trade-Related Aspects of Intellectual Property Rights).⁸² Er zieht daraus den Schluss, dass das internationale Handelsrecht angloamerikanischer Prägung das Persönlichkeitsrecht des Urhebers wie es traditionell in Europa gesehen wurde zunehmend aushöhlt. Dies ist auch die Folge aus der forcierten Umsetzung der EU- Richtlinie 2001/29/EG welche die Rechtsvorschriften, insbesondere das Urheber- und Patentrecht an die neuen Gegebenheiten der globalen Informationsgesellschaft „harmonisieren“ sollte.⁸³ Die großen Wellen die sich aus der Folge etwa um den § 52 a UrhG entwickelten wurden als der Untergang der Verlage gesehen. Wie man heute weiß sind seit Jahren gerade Verlage wie Elsevier mit Gewinnmargen von über 300% die Gewinner dieser Informationswirtschafts- „Krise“. Es wurde ein Art Modell entwickelt sich die Ergebnisse öffentlich finanzierter Forschung mit den von Bibliotheken erarbeiteten Metadaten anzueignen und wieder an Bibliotheken zurück zu verkaufen, ein eigentlich unzumutbares Vorgehen. Hier setzen auch Lohmeier und Mittelbach an, indem sie fordern dass Bibliotheken sich dem „Bündniszwang“ entziehen sollen und die Macht über die eigenen Daten und Dienstleistungen zurückgewinnen und dies auch mithilfe von OSS erreichen könnten.⁸⁴ Gerade in der Zeit Anfang der 2000er Jahre entstehen dann auch mit der Berliner Erklärung zu Open Access (2003) Tendenzen zur Benutzung freier Strukturen beim Publizieren wissenschaftlicher Forschung, was die Benutzung freier Software mit einschließt⁸⁵ und in dem schon beschriebenen Erfolg freier Repository- Software für Bibliotheken und Archive gemündet hat.

Das Haftungs- und Gewährleistungsrecht wird in den Open- Source- Lizenzen ausgeschlossen, die Benutzung von OSS geschieht damit „auf eigene Gefahr“. Das Marken-

⁸² Vgl. KUHLEN, Rainer. Wie öffentlich soll Wissen für Wissenschaft und Unterricht sein?. *Anmerkungen zum Urheberrecht in der Informationsgesellschaft*. In: EIBL, Maximilian. WOLFF, Christian (Hrsg.); WOMSER-HACKER, Christa (Hrsg.): *Designing Information Systems: Festschrift für Jürgen Krause*. Konstanz: UVK Verlagsgesellschaft mbH, 2005, 221. Jg., S.27-45.

⁸³ Vgl. c.europa.eu, 2016, Urheberrecht in der Informationsgesellschaft - Europäische Kommission. [online]. 2016. [Abgerufen 20 January 2016]. Verfügbar unter: http://ec.europa.eu/internal_market/copyright/copyright-infso/index_de.htm#maincontentSec1

⁸⁴ Vgl. Lohmeier 2014

⁸⁵ Vgl. Kuhlen 2005

recht erlaubt den Rechteinhabern von OSS- Produkten die Absicherung von Namen und Logos was die Einstufung als freie Software nicht aufhebt.

3.2 Softwarelizenzen

In § 12 und § 31 UrhG ist definiert dass der Autor einer Software diese mithilfe einer Lizenz publizieren darf, in der wiederum die Einräumung von Nutzungsrechten an den Schutzrechten geregelt ist. Es gibt momentan über 70 Open- Source- Lizenzen⁸⁶, von denen folgende am häufigsten verwendet werden: Berkeley Software Distribution (BSD), GNU General Public License (GPL), Eclipse Public License (EPL), GNU Library/Lesser General Public License (LGPL), Mozilla Public License (MPL), MIT License (MIT) und Apache License (APL). Die vier Grundprinzipien freier Software von Stallmann haben in der GPL, von der es zwei momentan gültige von insgesamt drei Versionen gibt, ihren unmittelbaren juristischen Ausdruck gefunden. Diese Lizenz wurde auch als der größte „juristische Hack“⁸⁷ aller Zeiten bezeichnet und war Inspiration für die Creative- Commons- Lizenzen. Die GPL sichert den freien Zugang zum Quellcode, die Weitergabe von Kopien, die Änderung des Programms und die Verbreitung des modifizierten Programms. Letzter Punkt ist nur gültig, wenn die Verbreitung der geänderten Software unter den gleichen Lizenzbedingungen geschieht⁸⁸, was eine mögliche reine Privatisierung der Software nicht erlaubt und die essentiellen vorigen Freiheiten bewahrt. Es wurde deswegen behauptet die weitere Verwendung von Software in einem funktionalen Zusammenhang mit solcherart lizenzierte Software sei nicht zulässig, was nicht der Wahrheit entspricht. Dies bezieht sich nur auf die aus dem originalen Quellcode abgeleitete Software und nur auf dessen Verbreitung. Als wahrscheinlich wichtigste OSS stehen der Linux- Kernel und GNU- Elemente wie der GNU- Compiler (GCC) und der Editor und Interpreter EMACS unter der GPL. Die GPL ist seit 2004 in mehreren Fällen in Deutschland für rechtsgültig erklärt worden.⁸⁹ Es gibt andere Lizenzen wie die BSD oder MPL die eine proprietäre Lizenzierung ausdrücklich erlauben, was es Microsoft auch in der Vergangenheit ermöglichte FreeBSD-Code in nicht unerheblichem Maß zu übernehmen und wieder als eigene Entwicklung zu verkaufen. Mit den Open- Source- Lizenzen wurde ein ausreichendes Rahmenwerk

⁸⁶ Vgl. Opensource.org, 2016, Licenses & Standards | Open Source Initiative. [online]. 2016. [Abgerufen 22 Januar 2016]. Verfügbar unter: <http://opensource.org/licenses>

⁸⁷ Vgl. Grassmuck 2004

⁸⁸ Dieses Prinzip ist auch als „Copyleft“ bekannt, was bedeutet dass sich die Bestimmungen unter allen Umständen auf alle abgeleiteten oder ergänzten Code weiter „vererben“.

⁸⁹ Zu der am weitesten verbreiteten GPL v2 vgl.: PICOT, Henriette. Die deutsche Rechtsprechung zur GNU General Public License. *Lutterbeck, Bernd; Bärwolff, Mathias; Gehring, RA (Hg.): Open Source Jahrbuch*, 2008, S. 233-244.

geschaffen dass es den Nutzern ermöglicht sich Software anzueignen selbst weiter zu entwickeln und mit anderen in einem rechtlich abgesicherten Raum zu teilen. Wie die Beispiele von KOHA und Quali OLE zeigen ist es unter diesen rechtlichen Rahmenbedingungen möglich bibliotheksspezifische Softwareentwicklung zu betreiben, ohne sich proprietären Einflussnahmen aussetzen zu müssen. So benutzt die Entwickler- Gemeinschaft um Quali OLE die Educational Community License 2.0 (ECL) einer abgeänderten Apache- Lizenz, welche festlegt dass erstens die Teile die Apache Software die verwendet werden zu benennen sind, zweitens der geänderte Code nicht an den Lizenzgeber zurückgehen muss und drittens eigene Softwareentwicklungen die Teile von Apache Software verwendet nicht unter der Apache License stehen muss. Allerdings wird das nächste große Upgrade auf Version 2.0 zur Affero GPL 3 wechseln, einer gerade für wissenschaftliche Bibliotheksverbände die als Host fungieren, interessanten Lizenzvariante, die Klauseln zum Hosting und eine Copyleft- Klausel beinhaltet.⁹⁰ Damit stellt die AGPL sicher dass in einem Netzwerk verbundene Benutzer einer dort zur Verfügung gestellten Software als Software as a Service (SaaS), einer immer häufiger anzutreffenden Nutzungsart, den Quellcode ohne Freiheitseinschränkungen nutzen können.

3.3 Patentrecht und FOSS- Lizenzen

Das Patentrecht erlaubt neben dem Plagiatsschutz auch den Schutz der eigentlichen Ideen und Grundsätze (eine einzelnes Element oder Schnittstelle) die dem Computerprogramm zugrunde liegen. Im europäischen Patentrecht und damit im deutschen Patentrecht ist die Patentierbarkeit softwarebezogener Erfindungen seit mehr als einem Jahrzehnt umstritten. Ein von der europäischen Kommission 2002 eingebrachter Vorschlag für eine Richtlinie über die Patentierbarkeit computerimplementierter Erfindungen ist vom europäischen Parlament abgelehnt worden. Die europäischen Patentämter verzeichnen in der Praxis aber weiterhin Anmeldungen dieser Art.⁹¹ Einer der Hauptintentionen des Patentrechts ist es technische Innovationen zu schützen. Aufgrund der kurzen Evolution von Software sind Patente aus Entwickler- Sicht aber ein Hindernis und bremsen einen möglichen Fortschritt aus. In der Praxis werden Patentanmeldungen vor allem in den USA dazu eingesetzt eine Monopolstellung zu sichern und die Konkurrenz auszuschalten oder zahlen zu lassen wo es nur geht. Die Gefahr Patentrechtsverletzungen zu begehen ist relativ hoch da auch so grundlegende Dinge wie Algorithmen

⁹⁰ Vgl. Borschnek 2015

⁹¹ Es gibt in Europa mehr als 70.000 eingetragene Softwarepatente die ohne eindeutige Rechtsgrundlage existieren.

patentfähig sind, unabhängig von Programmiersprache und Code der diese implementiert.⁹² Das United States Patent and Trademark Office (USPTO) listet momentan 922.565 Patentanmeldungen die direkt oder indirekt mit Software in Verbindung stehen.⁹³ Da das Patentrecht dem Territorialrecht angehört, sind nur international oder auf EU- Ebene angemeldete Patente auch in Deutschland wirksam, welches aber wiederum aufgrund der bereits erwähnten unklaren Rechtslage zu softwarebezogenen Erfindungen abgeschwächt wird.

Als mögliche Konflikte Bezug auf das Patentrecht und OSS sind weiter zu nennen⁹⁴:

1. Anmeldung eines Patentrechts Dritter auf bereits in der Open- Source- Community erbrachter Softwareentwicklung die als „Stand der Technik“ gilt aber den Patentämtern aufgrund fehlender Dokumentation unbekannt ist und deshalb als „neue“ Erfindung behandelt wird.
2. Patentrechtsverletzungen können sehr teuer werden und kleinere Open- Source- Entwicklergemeinschaften (Open- Source- Kernentwicklerteams bestehen nicht selten aus ein oder zwei Personen), damit sind auch bibliotheksinterne Entwicklerteams betroffen, können sich oft genug einen rechtlichen Schutz dagegen, etwa umfängliche, durch Anwälte gestützte Patentrechtsprüfungen, nicht leisten. Die Menge von Patentanmeldungen steigt der trotz unsicheren Rechtsgültigkeit weiter so dass es fast unmöglich ist alle potentiellen Rechtsverletzungen / (zumindest in den USA) zu prüfen.
3. Einzelne Entwickler innerhalb der freien Softwareentwicklung melden selbst Patente an und erreichen eine striktere Lizenzierung weiterentwickelten Codes.

Der unter Punkt 1 genannte Konflikt lässt sich durch eine ausführliche Dokumentation aller Prozesse innerhalb eines Software- Projektes vermeiden. Hier kann die transparente Nutzung eines Concurrent Version Systems⁹⁵ (CVS), also der internetbasierten Sourcecode- Verwaltung sowie die Veröffentlichung der Dokumentation (Handbücher, Leitfaden etc.) unter Creative- Commons- Lizenzen hilfreich sein. Für den zweiten Punkt ist eine potentielle Eingrenzung des Risikos nur bedingt möglich, wird aber

⁹² Eine sehr klare Darstellung zum Patent- und Urheberrecht im Kontext von OSS findet sich bei Jaeger, vgl.: JAEGER, Till; SCHULZ, Carsten. Gutachten zu ausgewählten rechtlichen Aspekten der Open Source Software. *Webseite: http://www.ifross.de/ifross_html/art47.pdf (05.01. 2016)*, 2005.

⁹³ Vgl.: Uspto.gov, 2016, United States Patent and Trademark Office. [online]. 2016. [Abgerufen 24 Januar 2016]. Verfügbar unter: <http://www.uspto.gov/>

⁹⁴ Jaeger 2005

⁹⁵ Auf Sourceforge.net befinden sich aktuell 696 OSS-Anwendungen für CVS- Systeme, vgl.: Sourceforge.net, 2016, cvs free download - SourceForge. [online]. 2016. [Abgerufen 24 January 2016]. Verfügbar unter: <http://sourceforge.net/directory/os:linux/?q=cvs>

durch die Bildung von Konsortien und der Aneignung juristischen Know- Hows abgemildert. Für den letzten Punkt gelten bestimmte Patentklauseln der verwendeten Lizenzen, die zwar ein Patent einräumen, den Verwendern der veränderten und patentierten Software aber wiederum urheberrechtlich- basierte Nutzung zugestehen müssen. In Deutschland beinhalten etwa die Bremer Lizenz für freie Softwarebibliotheken und die Deutsche Freie Software Lizenz entsprechende Klauseln.⁹⁶

Es ist bisher kein Patent- Rechtsstreit OSS betreffend in Deutschland bekannt.⁹⁷ Für das Markenrecht gilt das allerdings nicht. Hier gab es den Fall des quelloffenen und unter der GNU General Public License (GPL) stehenden Online- Shop- Systems xt:Commerce. Ein Anbieter veröffentlichte Software in modifizierter Form unter dem gleichen Namen. Hier entschied das Gericht das die GPL nicht die Vergabe und Weiterverbreitung eines Warenzeichens impliziert oder gestattet, aber die eigentliche Verbreitung der geänderten Software ausdrücklich erlaubt sofern der Bezug zum originalen Programm klar dokumentiert ist: „Nevertheless, the court pointed out that the lawful distribution of the free software remains possible, be it by choosing a different brand/trademark or by only referring to the fact that the modified software developed by the user is a modified version and compatible with the original “xt:Commerce” software.“⁹⁸ Da der Anbieter beides nicht tat wurde ihm die Nutzung des Namens verboten.

Es wurde angenommen dass Copyleft- Bestimmungen die in FOSS- Lizenzen zu finden sind gegen deutsches Wettbewerbsrecht verstoßen, was sich nicht bewahrheitet hat. In der Praxis unterstützen entsprechende Klauseln sowohl die Verbreitung derselben Software durch unterschiedliche Anbieter als auch funktionell ähnlicher Software durch andere Anbieter (siehe den Standard der OSI). Die Rechtslage von FOSS- Lizenzen im Zusammenhang mit dem deutschen öffentlichen Vergaberecht, und damit die Benutzung der Lizenzen in öffentlich finanzierten Bibliotheken, ist aktuell nicht eindeutig geklärt.⁹⁹ Es gelten die schon genannten Bestimmungen, Abschnitte und Gesetze im Urheberrecht.

⁹⁶ Jaeger 2005

⁹⁷ Vgl.: PEUKERT, Alexander; KÖNIG, Dominik. License Contracts, Free Software and Creative Commons in Germany. In: *Free and Open Source Software (FOSS) and other Alternative License Models*. Springer International Publishing, 2016. S. 201-226.

⁹⁸ Ebd., S.222

⁹⁹ Ebd.

Einer der berühmteren patentrechtlichen Streitfälle mit Bezug auf FOSS- Lizenzen in Amerika ist bis heute der Angriff von Microsoft und Apple auf das linuxbasierte Android. Die beiden Software- Giganten besitzen mehrere tausend Patente und sind sehr daran interessiert diese auch geltend zu machen. Microsoft hält insgesamt 310 Android- Patente. Samsung zahlte deswegen im Jahr 2013 ca. 1 Milliarde \$ an Lizenzgebühren. Allerdings war Microsoft in den Folgejahren weniger erfolgreich mit der Geltendmachung, so gelten 63,6 % der Ansprüche an diverse andere Anbieter von Android- Geräten mittlerweile als gescheitert, was auch für die diversen Versuche durch Apple gilt.¹⁰⁰ In einem markenrechtlichen Streitfall um Linux wurde durch Linus Torvalds, dem Entwickler des Linux- Kernels, und mehrere Linux- Organisationen eine Petition beim USPTO eingereicht und schließlich die Verwendung als Markenname durch den Anmelder nicht gestattet.¹⁰¹ Torvalds ist Eigentümer des Warenzeichens Linux und regelt mithilfe des Linux Mark Institutes über Sub- Lizenzen die Verwendung des Markennamens.¹⁰² Ein weiterer Streitfall im Markenrecht diesmal auch mit Bezug zu Bibliothekssoftware war der Streit um das BMS/ILS KOHA. Hier versuchte ein privatwirtschaftlicher amerikanischer Service- Anbieter für Bibliotheken, PTFS/LibLime, sich den Namen des von einer neuseeländischen Bibliotheks- Community entwickelten KOHA schützen zu lassen, was zunächst auch gelang. PTFS/LibLime eignete sich den quelloffenen Code von KOHA für eine abgeleitete Software an, vertrieb sie unter demselben Namen und erwarb die Nutzungsrechte der vorher von der freien Community verwalteten Repositorien, die Webseite, die E-Mail- Verteiler und kaufte die Domain koha.org.¹⁰³ Damit wurde versucht das bisher älteste freie Software- Projekt in der Bibliothekswelt proprietär zu machen. 2013 wurde der Namens- Streit beendet und alle Weiterentwicklungen am Code müssen von PTFS/LibLime an die Community zurückgegeben werden. Dennoch hat die Firma sich mit LibLime Academic Koha, das für ein Konsortium aus Informationseinrichtungen entwickelt wird einen eigenen Fork¹⁰⁴ geschaffen. Hier muss man einen finanziellen Beitrag leisten um Teil des Konsortiums

¹⁰⁰ Vgl. Vaughan-Nichols, Steven, 2014, Biggest patent win ever? Microsoft's billion dollar a year Samsung deal | ZDNet. *ZDNet* [online]. 2014. [Abgerufen 25 Januar 2016]. Verfügbar unter: <http://www.zdnet.com/article/biggest-patent-win-ever-microsofts-billion-dollar-a-year-samsung-deal/>

¹⁰¹ Vgl. MAGGS, Peter B. The Uncertain Legal Status of Free and Open Source Software in the United States. In: *Free and Open Source Software (FOSS) and other Alternative License Models*. Springer International Publishing, 2016. S. 477-493.

¹⁰² Vgl.: Linuxfoundation.org, 2016, Linux Trademark Institute. [online]. 2016. [Abgerufen 25 Januar 2016]. Verfügbar unter: <http://www.linuxfoundation.org/programs/legal/trademark>

¹⁰³ Für diesen und den folgenden Abschnitt vgl: Oss-watch.ac.uk, 2013, Koha: a case study in open source project ownership. [online]. 2013. [Abgerufen 25 Januar 2016]. Verfügbar unter: <http://oss-watch.ac.uk/resources/cs-koha>

¹⁰⁴ Als Fork werden Folgeprojekte aus einem Urprojekt verstanden, die die gleiche Codebasis haben.

zu werden und den Code einzusehen oder verändern zu dürfen. Da der Name weiterhin für Verwirrung sorgt hat PTFS/LibLime ein neues Softwareprodukt namens Bibliovation angekündigt, von dem vermutet wird, dass es auf großen Teilen von KOHA- Code aufbaut. Das immer noch durch die freie Community betreute KOHA ist bis heute eines der global am häufigsten eingesetzten freien Systeme in öffentlichen Bibliotheken. An diesem Beispiel zeigt sich wie erfolgreich auch ein kleiner Verbund auf dem juristischen Parkett agieren kann. Im amerikanischen Wettbewerbsrecht wurden Klagen und Beschwerden dass die FOSS- Lizenzen den freien Wettbewerb einschränken bisher zurückgewiesen. Hier wurde argumentiert die Garantie der Kostenfreiheit und der Weiterverbreitung durch die Copyleft- Klauseln verhindere eine Preissteigerung durch Anbieter nach Ausschalten der Konkurrenz.¹⁰⁵ Für die Verwendung von FOSS- Lizenzen nach dem öffentlichen Vergaberecht hat jeder Bundestaat eigene Regelungen für Land- und Gemeindeebene, die eine genaue Erhebungen der anfallenden Kosten für Installation, Support und Wartung eventuell einzusetzender OSS erfordern.

Da die Regelungen zum geistigen Eigentum OSS schützen befindet sich die Copyleft- Bewegung in einem Dilemma, denn zusehends werden die Gesetze des IPR als innovationsfeindlich gesehen. Die Intention hinter der Bewegung ist aber gerade Innovation zu fördern und Informationen ohne Restriktionen so weit wie möglich zu teilen: „It wants to provide maximum freedom for innovation, but also maximum information- sharing, but it can only do so in a situation where there is strong copyright protection, but weak patent protection.“¹⁰⁶ Es gilt also das Urheberrecht so zu verändern dass Innovation weiter möglich bleibt ohne es so einzuschränken dass OSS seiner wichtigsten Funktionen beraubt wird. Gerade hier könnten sich Bibliotheken als Agenten der Openness, vorausgesetzt man versteht ihre Funktion als solche, für die Verwendung freier Software einsetzen.

4 Problematik II – Politik, Philosophie und soziale Normen

„In jedem Fall ist von einer selbst programmierten Individuallösung abzuraten.“¹⁰⁷

¹⁰⁵ Vgl. Maggs 2016

¹⁰⁶ Ebd., S.492.

¹⁰⁷ Aus dem Praxishandbuch der deutschen katholischen Büchereifachverbände zum Thema Bibliothekssoftware, vgl. Buecherei-praxishandbuch.de, 2016, Bücherei Praxishandbuch: Bibliothekssoftware. [online]. 2016. [Abgerufen 27 Januar 2016]. Verfügbar unter: <http://www.buecherei-praxishandbuch.de/index.php?id=51>

Dieses Zitat aus einem Handbuch für katholische Bibliothekare zeigt die weiter verbreitete Ängstlichkeit wenn es darum geht die IT selbstständig zu verwalten. Dies mutet etwas seltsam wenn man sich klar macht dass die Biblioteca Apostolica Vaticana in Rom zu großen Teilen OSS erfolgreich implementiert hat wie CentOS oder Apache Hadoop und mit anderen wissenschaftlichen Bibliotheken regelmäßig Software- Lösungen tauscht und weiter aufbaut.¹⁰⁸ Ohne einer offiziellen Richtlinie zu folgen wird die Langzeitarchivierung unter Nutzung des offenen Dateiformats Flexible Image Transport System (FITS) vorangetrieben, ein Format dass 3-D Visualisierungen unterstützt, halbjährliche Updates erfährt, 64- Bit- Architekturen unterstützt und unbegrenzte Dateigrößen erlaubt (proprietäre Formate haben oft eine Begrenzung von wenigen GB). Die Bibliothekare des Vatikans nutzen einfach das vorhandene Wissen der Open-Source- Gemeinschaft um ihre lokalen Anforderungen digitaler Langzeitarchivierung zu meistern, ohne auf Anweisungen oder Richtlinien einer höheren Instanz zu warten. Dies zeigt neben der genannten Ängstlichkeit das zweite Problem, nämlich das Ignorieren nationaler Strategien und Richtlinien zum Einsatz von OSS in öffentlichen Strukturen, die durchaus existieren, was im Fall der Vatikanbibliothek aufgrund der isolierten Struktur der Vatikanstadt unproblematisch, aber in einem Staatenverbund wie der EU konfliktreich ist.

Ein funktionierendes Beispiel in vielen Ländern ist die digitale Langzeitarchivierung die auch durch genannte Charta in das Bewußtsein der Entscheidungsträger in Politik und Informationseinrichtungen gerückt ist. Programme wie das National Digital Information Infrastructure and Preservation Program in den USA oder das Gesamtkonzept der Kommission für die Informationsinfrastruktur¹⁰⁹ (KII) in Deutschland haben hierzu Vorgaben geliefert. In den nächsten Abschnitten wird nach einer kurzen Betrachtung der politischen Sichtweise auf das Thema Open Source in öffentlichen Strukturen auf die Ähnlichkeit der in der Open- Source- Community üblichen Normen und Philosophie mit dem bibliothekarischen Verständnis der Informationsdienstleistung eingegangen.

¹⁰⁸ Ein kurze Beschreibung und Volltextverlinkung zu einem Vortrag über die Software im Vatikan findet sich auf den Seiten der Joinup, einer Plattform der europäischen Kommission, vgl.: Joinup.ec.europa.eu, 2016, Vatican library: open source for long-term preservation | Joinup. [online]. 2016. [Abgerufen 27 January 2016]. Verfügbar unter: <https://joinup.ec.europa.eu/community/osor/news/vatican-library-open-source-long-term-preservation>

¹⁰⁹ Das 2011 erschienene Gesamtkonzept gilt immer noch als wesentlicher Orientierungspunkt, da die wichtigsten wissenschaftlichen Einrichtungen Deutschlands gemeinsam daran gearbeitet haben, was keine Selbstverständlichkeit ist.

4.1 Politische Statements – Die EU und Deutschland

Obwohl das Gesamtkonzept der KII einige Open- Source- Anwendungen zur Langzeitarchivierung nennt die man benutzen kann, gibt es keine ausformulierte OSS- Strategie für die behandelten Institutionen, das gleiche gilt für Publikationen der DFG. Grundsätzlich wird aber die Entwicklung von Software als förderungswürdig betrachtet, so heißt es in einem Merkblatt der DFG zum Förder- Programm „Werkzeuge und Verfahren des wissenschaftlichen Informationsmanagements“: „Im Mittelpunkt stehen dabei Projekte zur praxistauglichen Entwicklung von Software, Standards und Organisationsformen für Informationsdienstleistungen an wissenschaftlichen Infrastruktureinrichtungen, insbesondere für die Recherche, den Zugriff und die Bereitstellung von Informationen und digitalen Medien.“¹¹⁰ Auch hier wird nicht explizit auf eine mögliche Open- Source- Strategie verwiesen. Das politische Grundgerüst für Open Source und offene Standards für die europäischen Länder wurde von der europäischen Kommission gebaut. Neelie Kroes, bis 2010 EU- Wettbewerbskommissarin und bis 2014 EU- Kommissarin für die digitale Agenda war eine der maßgeblichen Befürworterinnen freier Software und offener Standards auf dem Weg zu einem E- Government dass den modernen Webstrukturen gerecht werden soll. In einer Rede von 2008 wies sie auf die Verpflichtung der europäischen Kommission hin keine geschlossenen Technologien mehr zu unterstützen und das der Zugang der Bürger einer Demokratie zu Informationen der Regierung nicht über die von nur einer Firma bereitgestellte Technik zu gewährleisten sei (Vendor- Lock- In), wenn gleichzeitig alternative Möglichkeiten offen stünden.¹¹¹ In späteren Äußerungen spricht sie von hunderten Millionen Euro die durch die Verhaftung an proprietäre Standards der europäischen Wirtschaft verloren gehen und dass kollaborative Softwareentwicklung die besseren Ergebnisse liefere. Die letzte veröffentlichte OSS- Strategie der europäischen Kommission die von 2014 – 2017 umgesetzt werden soll spiegelt diese Ansichten wider.¹¹² Neben dem immer zu präferierenden Einsatz von OSS soll auch mehr Klarheit im Urheberrecht und den Lizenzmodellen sowie gleiche Bedingungen bei Beschaffung und Teilhabe geschaffen werden. Im Zusammenspiel mit proprietärer Software können auch gemischte IT- Strukturen ge-

¹¹⁰ Die Merkblätter sind über die Homepage verfügbar, vgl.: Dfg.de, 2016, DFG - Deutsche Forschungsgemeinschaft - Formulare und Merkblätter / Gesamtliste. [online]. 2016. [Abgerufen 27 Januar 2016]. Verfügbar unter: http://www.dfg.de/foerderung/formulare_merkblaetter/index.jsp

¹¹¹ Vgl.: KROES, Neelie. Being open about standards. *OpenForum Europe. Brussel. Verfügbar unter: <http://europa.eu/rapid/pressReleasesAction.do>*, 2008.

¹¹² Einen kurzen Abriss der Geschichte der OSS- Strategien der EK seit dem ersten Entwurf findet man unter: Ec.europa.eu, 2016, Open Source Strategy in the European Commission: History - DIGIT. [online]. 2016. [Abgerufen 27 Januar 2016]. Verfügbar unter: http://ec.europa.eu/dgs/informatics/oss_tech/history/history_en.htm

fördert werden sofern faire Wettbewerbsbedingungen für beide Entwicklungsarten gesichert bleiben. Wie sieht nun die lokale Umsetzung, sprich die Praxis in den einzelnen EU- Staaten und damit Deutschland aus? Eine Tendenz wie flexibel sich die Länder neuen Informationinfrastrukturen im Bereich E- Government- Entwicklungen und damit offenen Standards öffnen zeigen die Rankings zwei bisher erschienener Studien der Vereinten Nationen aus den Jahren 2012 und 2014.¹¹³ Es wird untersucht wie effizient Regierungen E- Government- Richtlinien und Strategien umsetzen. Vergeben wird das Ranking nach dem E- Government Development Index (EGDI), der sich aus dem Online- Service- Index (OSI), Telecommunication- Infrastructure- Index (TII) und dem Human- Capital- Index zusammensetzt (HCI) die vielfältig mit anderen Indikatoren wie Bildungsstand und Einkommen kombiniert werden. Der OSI beschreibt dabei die Bandbreite und die Qualität der angebotenen amtlichen Online- Services, der TII den Entwicklungsstatus der physischen Infrastruktur alle das Netz betreffenden Technologien und der HCI den Qualifikations- und Motivationsgrad der Regierungsmitarbeiter. Der Index hat die Formel $EGDI = 1/3 (OSI_{normalisiert} + HCI_{normalisiert} + TII_{normalisiert})$ ¹¹⁴. Demnach liegt Deutschland auf Platz 22 (-11 seit 2012) mit dem restlichen Europa auf den letzten Plätzen, die USA auf Platz 7 (-2 seit 2012) unter den ersten Zehn in der Gruppe von 27 Ländern die einen überdurchschnittlichen EGDI- Index von 0.8368 bei einem Weltdurchschnitt von 0.4712 aufweisen. Die Umsetzung der Strategie der EK auf der Ebene regionaler Verwaltungen sieht im Kontext OSS sehr widersprüchlich aus. Öffentliche Stadtverwaltungen, darunter Wien, München oder Freiburg, haben seit Mitte der 2000er Jahre begonnen offene Standards wie das Open Document Format (ODF) und OSS zu integrieren sind aber teilweise wieder zu proprietären Systemen zurückgekehrt.¹¹⁵ Die Stadt München wurde von Windows wiederholt unter Druck gesetzt ihre Infrastruktur wieder auf dessen System zurück zu setzen. Die Motivation der Stadt unabhängig von einem einzigen proprietären Anbieter zu sein dem die Belange der Aktionäre wichtiger als die Belange der Bürger sind, hat bis heute Bestand. Die Selbstbestimmung ist wesentlicher Faktor bei der Adaption von OSS in

¹¹³ Vgl. UN, E. Government Survey. 2014. *E-government for the Future We Want* - http://unpan3.un.org/egovkb/Portals/egovkb/Documents/un/2014-Survey/E-Gov_Complete_Survey-2014.pdf, Abgerufen Januar, 2016, 28. Jg.

¹¹⁴ Vgl.: Publicadministration.un.org, 2016, Methodology. [online]. 2016. [Abgerufen 28 Januar 2016]. Verfügbar unter: <https://publicadministration.un.org/egovkb/en-us/About/Methodology>

¹¹⁵ Ein Beispiel hierfür ist die Stadt Freiburg. Als wesentlichen Motivationsgrund der Rückkehr wurden Kompatibilitätsprobleme beim Austausch von Textdokumenten mit anderen Stadtverwaltungen genannt, vgl.: Joinup.ec.europa.eu, 2016, 'Freiburg's frustration with IT caused anger over free software' | Joinup. [online]. 2016. [Abgerufen 27 Januar 2016]. Verfügbar unter: <https://joinup.ec.europa.eu/news/freiburgs-frustration-it-caused-anger-over-free-software>

Verwaltungsstrukturen.¹¹⁶ Das auswärtige Amt, eines der berühmtesten Beispiele für die Adaption von Open Source in Deutschland hat ebenso die meisten Desktop- PC's 2014 wieder auf MS Windows umgestellt, nachdem 2008 mehr als die Hälfte der Rechner in den Botschaften und Konsulaten weltweit zu linuxbasierten Desktop- Systemen migrierte. Als Gründe wurden die angeblich bessere Wirtschaftlichkeit des proprietären Modells und fehlende Nutzerfreundlichkeit genannt was beides aber nicht als erwiesen gilt. Die fehlende Möglichkeit der Mitarbeiter lokal installierte Programme selbständig zu aktualisieren was natürlich Kompatibilitätsprobleme in Netzwerk- und Hardwarekomponenten entstehen lässt hat nichts mit freier Software oder oder fehlenden Admin- Rechten zu tun, die Hauptargumente der Kritiker, sondern mit den lokalen Sicherheitsvorschriften. Die Partei der Grünen arbeitet mit der Free Software Foundation Europe (FSFE) zusammen um den Druck auf die Bundesregierung zu steigern eine verbindliche und klare Strategie zur vermehrten Benutzung Freier Software umzusetzen,¹¹⁷ wie es die Europäische Kommission seit über einem Jahrzehnt aktiv betreibt. Es bleibt festzuhalten dass im europäischen Vergleich Deutschland bei der Implementierung freier Software- Strategien weit zurückliegt.¹¹⁸ Trotz der Bekenntnis zu offenen Standards geht das Kompetenzzentrum Open Source der Bundesstelle für Informationstechnik (BIT) in seinen Publikationen bis auf einen Migrationsleitfaden der einen sehr brauchbaren Regelkatalog enthält kaum auf OSS ein. Eine kohärente Strategie ist nicht zu finden.

Die europäische Kommission hat in ihrer Prioritätensetzung vom Januar 2016 den digitalen Binnenmarkt als reformbedürftig erkannt. Neben einer erneuten Harmonisierung des europäischen Urheberrechts sollen eine Verbesserung des Zugangs zu digitalen Waren und die Interoperabilität der Technologien im Sektor des Cloud- Computing und das Internet der Dinge.

¹¹⁶ Dies ist neben der Notwendigkeit von Innovation einer der Hauptgründe für die Umstellung, vgl. CASSELL, Mark. Why governments innovate: Adoption and implementation of open source software by four European cities. *International public management Journal*, 2008, 11. Jg., Nr. 2, S. 193-213.

¹¹⁷ Vgl. <http://www.heise.de/open/meldung/Auswaertiges-Amt-hat-sich-weitgehend-von-Linux-verabschiedet-2595478.html>. Eine Anfrage der Grünen zu den spezifischen Gründen, Kosten und Details der Rückkehr zu Windows und die Antwort der Bundesregierung sind einsehbar unter: [Dipbt.bundestag.de](http://dipbt.bundestag.de), 2016, DIP21 Extrakt. [online]. 2016. [Abgerufen 28. Januar 2016]. Verfügbar unter: <http://dipbt.bundestag.de/extrakt/ba/WP18/653/65399.html>

¹¹⁸ Der Journalist Gijs Hillenius, der unter anderem für Open – Source Plattform der EK, Joinup, arbeitet spricht in diesem Zusammenhang von „German Angst“, vgl.: Gijs Hillenius, Markus Feilner, 2016, Europas Behörden vertrauen zunehmend auf Open-S... - Seite 5 » Linux-Magazin. *Linux-Magazin* [online]. 2016. [Abgerufen 28. Januar 2016]. Verfügbar: <http://www.linux-magazin.de/Ausgaben/2013/11/Open-Source-in-Europa/%28offset%29/8>

4.2 Politische Statements – USA

Wie viele andere Regierungen der Welt haben auch die USA die aktuelle Transformation der IT- Infrastruktur zu Modularisierung, Nachhaltigkeit und mehr Interoperabilität erkannt. Die zunehmend geringen Budgets führen auch hier zu einer Zielsetzung der Vermeidung eines Vendor- Lock- In wie ihn auch Neelie Kroes für die EU anempfohlen hatte. In einem Memorandum von 2012 spricht Barack Obama über die Neuausrichtung der IT- Strategie für das 21. Jahrhundert. Die Implementierung der Strategie: „Digital Government: Building a 21st Century Platform to Better Serve the American People“ sollte innerhalb von 12 Monaten nach Veröffentlichung des Memos erfolgen. Nach spätestens 90 Tagen sollte über eine zentrale Webseite der Regierung zur digitalen Strategie der Fortschritt der Umsetzung in den Ämtern dokumentiert werden.¹¹⁹ Die formulierte Strategie weist eine eindeutige Ausrichtung auf Open Source auf und beinhaltet neben der Festsetzung Open Data und Open Content als Standard zu etablieren auch die Aufforderung der Partizipation aller Behörden an Open- Source- Projekten. Das weisse Haus selbst ist auf GitHub mit eigenen Repositories vertreten und als Contributor aktiv.¹²⁰ Etwas derart Progressives sucht man bei der deutschen Bundesregierung vergebens. Zwei weitere Initiativen der unabhängigen Behörde General Service Administration sollen helfen die Zielvereinbarungen in Regierungsbehörden zu verankern: 18F¹²¹ und U.S. Digital Service¹²². Beide Teams haben ein klare Agenda zum Einsatz von Open Source unter den Bedingungen FOSS in allen Projekten zu nutzen und den Quellcode öffentlich zu machen. Veröffentlichte Checklisten und Schlüsselfragen sollen den Behörden helfen die Anpassung ihrer jeweilige Struktur der digitalen Service-Angebote an den Strategiezielen effektiver zu gestalten.

Probleme bei der Implementierung auf lokaler Ebene sind in Amerika weniger auf den politischen Rückhalt wie dies in Deutschland der Fall ist zurückzuführen. Einer der Gründe die genannt werden ist das fehlende Wissen der Entscheidungsträger in Behör-

¹¹⁹ Die Pflicht zur Veröffentlichung ist in der Vollzugsanordnung 13571 („Streamlining Service Delivery and Improving Customer Service“) festgelegt. Das Memorandum und die formulierte Strategie sind über die Webseite des weissen Hauses verfügbar, vgl.: whitehouse.gov, 2015, Developers. [online]. 2015. [Abgerufen 28 Januar 2016]. Verfügbar unter: <https://www.whitehouse.gov/developers>

¹²⁰ GitHub ist das größte Repositorium freier Softwareprojekte. Im Profil des weissen Hauses finden sich Projekte zu API's, Petitionen oder Drupal- Modulen, vgl.: GitHub, 2016, The White House. [online]. 2016. [Abgerufen 28 Januar 2016]. Verfügbar unter: <https://github.com/whitehouse>

¹²¹ Das Team entspricht in Zusammensetzung und Ausrichtung einem unabhängigen Consulting- Unternehmen, vgl.: Us, Join, 2016, 18F Digital Services Delivery. *18F Digital Services Delivery* [online]. 2016. [Abgerufen 28 Januar 2016]. Verfügbar unter: <https://18f.gsa.gov/>

¹²² Im Gegensatz zu 18F ist das Team Teil des Executive Office und damit des weissen Hauses.

den um die Eigenarten von Software wenn es um die Behebung von Fehlern (bugfixes) geht. Dabei spielt die Tatsache dass die höchsten Kosten bei OSS bei der Wartung entstehen. Die Praxis, Kosten dadurch zu reduzieren dass man sich die Fehlerbehebung durch das frühzeitige Teilen der betreffenden Quellcodestellen und die Kommunikation mit der Community holt hat sich vielfach noch nicht durchgesetzt.¹²³ Weitere Faktoren sind falsche Marktanalysen durch Manager ohne Technik- Affinität (ältere Mitarbeiter), Angst vor Veränderung, Angst vor Migrationskosten oder das fehlende Wissen dass es sich bei OSS in den USA in den meisten Fällen um kommerzielle Software handelt. Laut gesetzlichen Bestimmungen ist jede Software als kommerziell zu betrachten die über eine Lizenzierung der Öffentlichkeit übergeben wird. Letzter Punkt erleichtert eigentlich die Adaption, da es einen festen Regelkatalog, ähnlich des Migrationsleitfadens des BIT in Deutschland gibt, der es Behörden erleichtert freie Software einzuordnen und zu nutzen. Bestimmte Regeln gehen soweit dass Mitarbeiter einer Regierungsbehörde gegen das Gesetz verstoßen wenn sie in einer zu dokumentierenden Marktanalyse OSS nicht in Betracht ziehen. Ein großes Problem sind eigenständige Forks, die auf Seiten der Vertragspartner mit der Motivation entstehen sich die exklusiven Rechte an der Software zu sichern und eine für die Behörde so spezielle Software zu entwickeln, dass diese. Die externen Vertragspartner die sich mit Abzweigung von der Hauptentwicklungslinie einer Software zukünftige Aufträge zur Anpassung und Wartung sichern, die bei Projektabsplaltungen zwangsläufig vermehrt auftreten, sind oft genug nicht an der politischen Agenda interessiert. Ein Problem dass für beide Länder evident ist.

Der größte Unterschied beider Länder liegt in dem Bekenntniss der amerikanischen Regierung den Prinzipien der Open- Source- Bewegung politischen Rückhalt zu geben was zum Abbau der immer noch beobachtbaren Unsicherheiten beiträgt. So heißt es auf der Webseite des weissen Hauses: „We believe in using and contributing back to open source software as a way of making it easier for the government to share data, improve tools and services, and return value to taxpayers.“ Eine prägnante Aussage dieser Art ist trotz der Ansätze seitens der europäischen Kommission von der Bundesregierung aktuell nicht zu erwarten.

¹²³ Für diesen Aspekt und die folgenden vgl.: Opensource.com, 2016, US government accelerating development and release of open source. [online]. 2016. [Abgerufen 29 Januar 2016]. Verfügbar unter: <https://opensource.com/government/14/4/interview-david-wheeler>

4.3 Philosophien und Normen – die Open- Source- Community und das bibliothekarische Verständnis der Informationsdienstleistung

„[...] if something doesn't fit, we adapt it, replace it, or rewrite it. Our vision doesn't change easily, it remains more or less the same, we're just constantly refining it.“¹²⁴

Bevor man Aussagen zur Philosophie und den Normen der Open- Source- Community trifft muss man sich das zugrunde liegende Modell freier Softwareentwicklung vergegenwärtigen. Der meistzitierte Text in diesem Zusammenhang stammt von Eric Raymond, einem der Gründer der OSI und Befürworter der kommerziellen Ausrichtung von OSS. Er vergleicht darin das klassische Modell mit einer Kathedrale und das freie Modell als Basar.¹²⁵ Während der Quelltext in ersterem nur selten und in großen Abständen veröffentlicht wird, werden auf dem Basar in kurzen Abständen aktualisierte und verbesserte Versionen des ursprünglichen Codes verteilt, was für mehr Innovation und regeren Handel sorgt. In einem Projekt das sich nach Open- Source- Prinzipien ausrichtet lassen sich in der Regel acht über verschiedene Kommunikationsprozesse verbundene Personengruppen ausmachen.¹²⁶

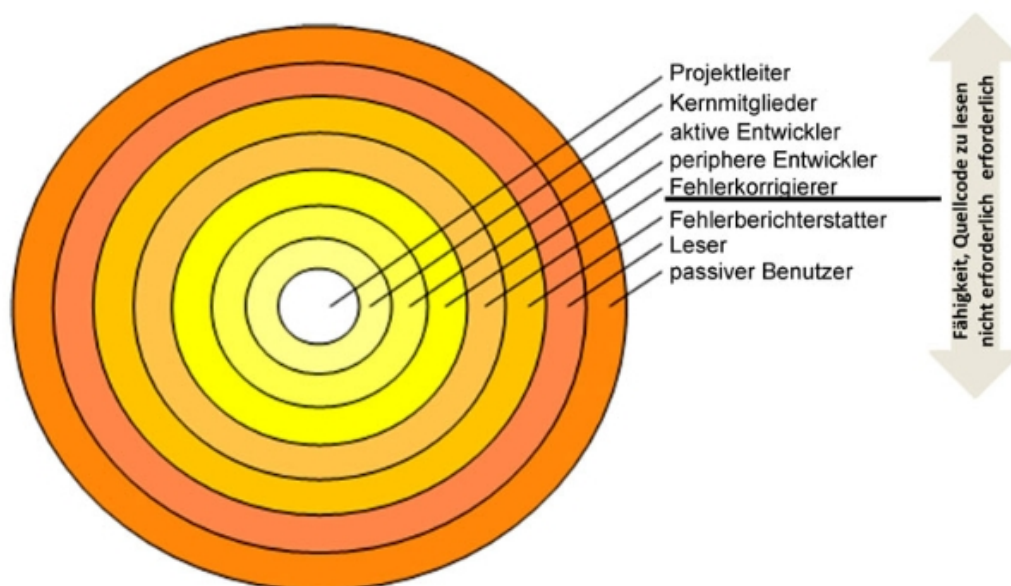


Abbildung 3: Das Schalenmodell eines FOSS-Projekts

Quelle: Kees 2015

¹²⁴ Dieses Zitat aus den Monatsnachrichten der Linux- Mint- Community, einem der erfolgreicheren Betriebssysteme für den Desktop, zeigt klar die Motivation sich ohne Unterlaß auf Neuerungen einzustellen, vgl.: The Linux Mint Blog, 2016, Monthly News - January 2016. [online]. 2016. [Abgerufen 29. Januar 2016]. Verfügbar unter: <http://blog.linuxmint.com/?p=2985>

¹²⁵ Obwohl dieses Gleichnis aufgrund der oft stark hierarchischen Strukturen in Open Source Projekten aus heutiger Sicht eher irritiert, gilt der Text als ideeller Meilenstein in der Entwicklungsgeschichte freier Software. Vgl.: RAYMOND, Eric. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 1999, 12. Jg., Nr. 3, S. 23-49.

¹²⁶ Kees 2015

Im Zentrum stehen eine oder mehrere Personen¹²⁷ (Projektleiter) die die Vision und Strategie vorgeben und die meisten Entscheidungen treffen. Die Kernmitglieder sind für die Koordination und das Qualitätsmanagement verantwortlich und nehmen bei Bedarf die Rolle des Leiters ein und oft selektieren und testen sie den Code der nächsten Gruppe von aktiven Entwicklern. Aktive Entwickler sind hauptverantwortlich für bugfixes und stellen in regelmäßigen Abständen große Abschnitte von Quellcode, oft Erweiterungsmodule, zur Verfügung. Periphere Entwickler tun dasselbe in größeren Abständen. Die nächste Gruppe der bugfixer sucht und behebt selbst erkannte Fehler oder übernimmt diese von den aktiven Entwicklern. Die Gruppe der bug-reporter sind selbstständige Prüfer die Fehler erkennen und diese weiter kommunizieren (oft werden diese gezielt als Tester von Beta- Versionen herangezogen), aber nicht in Eigenregie lösen. Die letzte Gruppe sind die aktiven und passiven Nutzer. Erstere lesen sich in die Tiefenstruktur des Codes ein und kommentieren die Funktionalität und Flexibilität der Programme in entsprechenden Foren und sind meist selbst Entwickler anderer Projekte, während passive Benutzer als reine „User“ gesehen werden die zwar auch kommentieren und berichten, aber dies in den allgemeineren Foren tun. Neben üblichen Mail- und IRC- Kanälen, Bug- Datenbanken und anderen Repositorien sind bis heute die oft nach den einzelnen Forks aufgeteilten Foren auf den Projektseiten das erste Mittel der Kommunikation.¹²⁸ Besonders häufig findet man derart Haupt- und Unterforen auf dem Feld der linuxbasierten Betriebssysteme für den Desktop.¹²⁹ Bei kleineren Projekten wird über und durch jeden Schalenring (siehe Abb. 3) kommuniziert, bei größeren wie dem Linux- Kernel mit bis zu tausend aktiven Entwicklern setzt eine stärkere Filterung ein. Torvalds kommuniziert nur mit dem nächsten Schalenring, also ca. 5-6 Kernmitgliedern, die alles was sie für den Kernel wichtig erachten zur Entscheidungsfindung vorlegen. Obwohl dieses Beispiel etwas dem angenommenen kollaborativen Charakter abträglich ist, werden vom Kernprogrammierer bis zu Endnutzer auch hier alle in Entwicklungsprozess einbezogen.

¹²⁷ Projekte wie der Linux- Kernel haben einen starken Personenbezug, in diesem Fall Linus Torvalds, dem das Charakteristikum eines „benevolent Dictators for Life“ zugeschrieben wird, einer in der Open- Source- Bewegung humoristischen Betitelung die aber letztendlich die Machtposition treffend beschreibt. Auch im proprietären Modell ist das Phänomen bekannt (Steve Jobs, Bill Gates etc.).

¹²⁸ Vgl: ETRICH, Matthias. Koordination und Kommunikation in Open-Source-Projekten. *Open Source Jahrbuch*, 2004, S. 179-192.

¹²⁹ Hier sind Unterteilungen nach jeweiliger Desktopumgebung üblich. So gibt es für das OS Manjaro neben dem von den Kernentwicklern bevorzugten XFCE zahllose Spin- Offs mit weniger verbreiteten Umgebungen wie Fluxbox oder i3 oder Mate die sich in den Unterforen wiederfinden und als Community- builds bezeichnet werden.

Persönliche Begegnungen sind auf den zahlreichen Messen, Symposien und sonstigen Veranstaltungen möglich. Dazu gehören neben eher traditionellen Treffen wie der vom Verlag O'Reilly organisierten OSCON¹³⁰ auch reine Hacker- Conventions wie die DEF CON. Die soziale Dynamik der Open- Source- Gemeinschaft definierte sich anfangs stark über ein gemeinsames Feindbild, dessen Vertreter noch 2004 folgend beschrieben wurden: „These enemies include anything and anyone which is perceived as prohibiting access, including copyrights, patents, and secret source codes, but also mechanisms that encourage dependence.“¹³¹ Das klingt den aktuellen Klagen vieler Bibliothekare über ineffektive Verlagsmodelle und der Forderung nach umfassenden Open Access sehr ähnlich.¹³² Was einer Hacker- Subkultur entsprach hat sich allerdings dem Mainstream angepasst und wie schon angedeutet in einen politischen (FSF) und kommerziellen (OSI) Teil gespalten. Nichtsdestotrotz hat sich eine offene Kultur direkter Kommunikation und des Teilens etabliert, die den offenen Quelltext als zentrales Informationsgut betrachtet dessen Verbreitung in alle Infrastrukturebenen der vernetzten Welt bereits stattgefunden hat und kontinuierlich bisher proprietäres Gebiet infiltriert. Bibliotheken haben im Gegensatz zur freien Softwareentwicklung einen explizit politischen Auftrag als Agenten des Art. 5 GG oder des Art. 19 der UN- Menschenrechtserklärung¹³³ und gelten als Garanten der demokratischen Partizipation an Information. Trotzdem müssen sie sich mehr und mehr Strategien ausdenken um der Politik ihre Relevanz zu beweisen. Freie Softwareentwickler haben es geschafft nach Prinzipien eine Gemeinschaft kollektiver Intelligenz und des Wissens zu etablieren die niemanden und nichts ausschließt, sondern jeden willkommen heißt der fähig ist etwas Konstruktives zum Projekt beizutragen, also Contributor zu werden. Wie das Schalenmodell zeigt muss man dafür keineswegs Code schreiben können. Grassmuck hat vor über zehn Jahren die Bibliotheken als Teil einer öffentlichen Wissensinfrastruktur gesehen die Werkzeuge wie Protokolle, Metadaten und Suchmaschinen mit den Wissensbeständen zu verknüpfen habe damit die informationelle Grundversorgung und die Schaffung einer Wissensallmende Wirklichkeit werden kann.¹³⁴ Und tatsächlich sind Bibliotheken auch immer wieder Vorreiter in bestimmten Aspekten der IuK- Technologien gewesen, gerade in Schaffung von offenen Standards zur Förderung der Interope-

¹³⁰ Vgl. [Conferences.oreilly.com](http://conferences.oreilly.com), 2016, O'Reilly Open Source Convention. [online]. 2016. [Abgerufen 29. Januar 2016]. Verfügbar unter: <http://conferences.oreilly.com/oscon>

¹³¹ Vgl.: LEHMANN, Frauke. FLOSS developers as a social formation. *First Monday*, 2004, 9. Jg., Nr. 11.

¹³² Vgl. Lohmeier 2015

¹³³ Vgl. BERGMANN, Julia; DANOWSKI, Patrick (Hg.). *Handbuch Bibliothek 2.0*. Walter de Gruyter, 2010.

¹³⁴ Vgl. Grassmuck 2004

rabilität wie etwa der Standardisierung der Katalogkarte Ende des 19. Jh. oder dessen Computerisierung durch MARC von Henriette Avram, eine der ersten Programmierinnen der Welt und späteren Bibliothekarin. Gleichzeitig sind diese ehemaligen Innovationen das heutige Problem moderner Bibliotheken. Die meisten Technologien, Standards und Protokolle die in und von Bibliotheken entwickelt oder benutzt wurden und werden erzeugen nur in einer innerbibliothekarischen Bezugswelt einen funktionalen Kontext. Während es vor dem WWW der ersten Generation noch sinnvoll schien ein rein auf Bibliotheken ausgerichtetes Datenaustauschformat zu entwickeln, ist im Zeitalter des ubiquitous computing und der immensen Metadatenproduktion von Konzernen wie Amazon etwas offeneres, flexibleres und nicht zuletzt wettbewerbsfähigeres gefragt, denn: „Daten sollen Beziehungen zwischen Objekten ausdrücken und maschinenlesbar vermitteln können, nicht mehr nur Objekte statisch und isoliert voneinander beschreiben.“¹³⁵

Kein Mensch außerhalb eines Bibliotheksbüros versteht was MARC ist aber jeder kann etwas mit HTML, HTTP oder zumindest einem Browser etwas anfangen, den Grundbausteinen des erfolgreichsten OSS-Projekts aller Zeiten. Berners-Lee entwickelte seine Idee genau wie Stallman und Torvalds aus Unzufriedenheit über den technischen Standard seiner Zeit und programmierte die Lösung einfach selbst. Was bei Stallman der Ärger über die fehlende Einsicht in den Quellcode eines Treibers für einen Xerox-Netzwerkdrucker war oder bei Torvalds das Fehlen eines passenden Betriebssystems für seine spezielle Hardware, war bei Berners-Lee die Unzufriedenheit mit der wissenschaftlichen Kommunikation beim CERN. Bis heute ist es möglich als interessierter Nutzer von Information oder gar Programmierer an diesen Projekten, seien es GNU, der Linux-Kernel oder die Webstandards des W3C, mitzuwirken. Die „Architektur der Partizipation“¹³⁶ wie sie durch freie Softwareentwickler als Entwicklungs- und Verbreitungsmodell von Wissen gelebt wird, entspricht in ihrem Kern, nämlich der Kommunikation und des Teilens von (Fach-) Information über alle Grenzen hinweg dem Selbstverständnis von Bibliotheken, wie es seit der frühneuzeitlichen Umdeutung existiert. „Bibliotheken als Garant von Offenheit sind quasi schon verpflichtet

¹³⁵ Vgl.: ALTENHÖNER, Reinhard. *Jenseits der Cloud: Metadaten- und Datenmanagement in der bibliothekarischen Infrastruktur Teil 1*. Bibliotheksdienst, 2015, 49. Jg., Nr. 7, S. 677-695.

¹³⁶ Ein Begriff der von Tim O'Reilly geprägt wurde, Kopf einer der wichtigsten Verlage für OSS-Literatur, vgl.: O'Reilly, Tim, 2016, *The Architecture of Participation* - O'Reilly Media. *Archive.oreilly.com* [online]. 2016. [Abgerufen 30. Januar 2016]. Verfügbar unter: http://archive.oreilly.com/pub/a/oreilly/tim/articles/architecture_of_participation.html

dazu, zumindest in den Kern-Tätigkeitsbereichen Open-Source einzusetzen und mit zu entwickeln und damit auch zu fördern.“¹³⁷

5 Problematik III – Der Code

Will man eine Übersicht über die Grundstruktur des Quellcodes gewinnen wie sie in Bibliothekssoftware existiert sind neben den Code- Repositorien auf GitHub einzelne Portale sehr hilfreich, in denen über die Transformation von Software in Bibliotheken in einem breiteren Kontext gesprochen wird und in denen Beiträge von Programmierern zu finden sind die in oder eng mit Bibliotheken zusammenarbeiten. Die folgenden Beispiele stammen aus Artikeln des OA- Journals Code4Lib, einer Zeitschrift zum Thema Programmieren und Code in Bibliotheken mit interdisziplinären Ansatz.¹³⁸ Es wurde bereits angemerkt das technisches Know- How in größerem Maß vorhanden sein muss um moderne Software an die lokalen Anforderungen von bibliothekarischen Arbeitsabläufen anzupassen. Dies kann zum einen über die Auslagerung an eine externe Software- oder Supportfirma geschehen oder in Eigenregie mit entsprechend geschultem Personal. Die Mindestanforderungen die auch schon heute in relevanten Studiengängen in den USA und Deutschland unterrichtet werden sind neben XML HTML, PHP, MySQL, SPARQL, JSON (Datenaustauschformat) und die Skriptsprache Perl. Gerade letztgenannte zeichnet sich durch Modularisierung und Plattformunabhängigkeit aus und gilt als einfach zu lernen. Es ist nicht ausreichend die Arbeit mit dem Code an eine Handvoll Systembibliothekare abzugeben: „Die Digitalisierung hält auf allen Ebenen unserer Gesellschaft unserer Welt derart mächtig Einzug, dass wir es uns nicht leisten können als Informationsanbieter nicht auch ganz substantielles Know-How aufzubauen. Das geht eigentliche nur wenn wir die Technologien verstehen.“¹³⁹ Folgende Beispiele sind aus den Jahren 2013 und 2015.

5.1 Beispiel I: Ein Suchindex aus LOD- Datensätzen für Solr

Linked Data ist seit mehr als 5 Jahren eines der Themen im neueren Aufgabenfeld für Bibliotheken. Vor allem große Häuser wie die Library of Congress und die deutsche Nationalbibliothek haben Initiativen zu LOD gestartet. Seit der Implementierung der Strategie des Bibliographic Framework for the digital Age (BIBFRAME) seitens der

¹³⁷ Vgl. Interview AI, Z. 136-139.

¹³⁸ „It is a volunteer-driven collective of hackers, designers, architects, curators, catalogers, artists and instigators from around the world, who largely work for and with libraries, archives and museums on technology "stuff." Vgl.: code4lib, 2011, About. [online]. 2011. [Abgerufen 30 Januar 2016]. Verfügbar unter: <http://code4lib.org/about>

¹³⁹ Vgl.: Interview Anhang I, Z. 62-65.

USA um endlich MARC21 und Z39.50 auszuwechseln scheint der Weg bibliographischer Metadaten aus dem Deep Web vorgezeichnet.¹⁴⁰ Das Online-Repository DataHub verzeichnet momentan 1.228 LOD-Datensätze¹⁴¹ (z.B. des B3Kat). Trotzdem bleibt die Implementierung semantisch angereicherter Daten eine Herausforderung, was man auch an der sehr langsam wachsenden Anzahl von SPARQL-Endpoints sieht. Nachfolgend werden Änderungen am Discovery-System VuFind beschrieben.¹⁴²

Für das folgende Beispiel sollte für die VuFind Suchmaschine Solr ein Suchindex für RDF-Daten erstellt werden um so die als LOD i vorliegenden Datensätze im Discoverysystem verfügbar zu machen. Solr bietet Volltextsuche, Ranking und Relevanz, Faceted Search, Filterung mit regulären Ausdrücken und benutzt über entsprechende Schnittstellen HTTP/XML und JSON zur Kommunikation. Ziel des von Hatop durchgeführten Projektes war es ein einfaches System zu entwickeln dass über SPARQL-Abfragen Daten aus LOD ziehen kann und über XSLT-Transformationen Solr-Indexdateien herstellen kann. Als Mittel der Umsetzung wurden ausschließlich semantische Technologien und vorhandene Standards verwendet. Wesentlicher Vorgang ist das Laden der RDF-Daten in den Triple-Store und der Zugriff über die API Schnittstelle des Stores. Dies wurde über Java und das Apache Jena-Framework realisiert.¹⁴³ Nachdem alle Daten in den Jena-Triple-Store geladen wurden folgten die Schritte der Aufzählung aller Einträge, das Erstellen eines Auszugs aus der Datenbank um zu sehen was der Triple-Store an Informationen über die Ressource hat (resource dump) und schließlich das Mapping und die Filterung der Ausgabe für die Indexierung. Letzter Schritt ist mit XSLT, die vorigen mit SPARQL durchführbar. Die SPARQL-Abfrage die zu der Aufzählung des gesamten im Store gespeicherten Inhalts führt hat die folgende Syntax:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
# general resource enumeration : every subject, but no blank nodes
select distinct ?subject
where {
  ?subject ?p ?o .
  FILTER (!isBlank(?subject)) .
}
ORDER BY DESC(?subject)
```

¹⁴⁰ Ein Artikel von Enis beschreibt anschaulich wie sich die Isolation bibliographischer Metadaten endgültig mit BIBFRAME überwinden lässt, vgl.: ENIS, Matt. Ending the invisible Library. *Library Journal*, 2015, 140. Jg., Nr. 3, S. 36-38.

¹⁴¹ Im Jahr 2013 waren es nur ca. 300, vgl.: Datahub.io, 2016, Search for a Dataset - the Datahub. [online]. 2016. [Abgerufen 30. Januar 2016]. Verfügbar unter: <https://datahub.io/dataset?tags=lod>

¹⁴² Für den gesamten folgenden Abschnitt vgl.: HATOP, Götz. Integrating Linked Data into Discovery. *Code4Lib journal*, 2013, 21. Jg., S. 67-90.

¹⁴³ Apache Jena ist OSS, vgl.: Jena.apache.org, 2016, Apache Jena -. [online]. 2016. [Abgerufen 30. Januar 2016]. Verfügbar unter: <http://jena.apache.org/>

Damit erreicht man die Ausgabe der Einträge aller vorhandenen Ressourcen bis auf die blank nodes (also Ressourcen im RDF- Graph ohne URI). Um einen Datenbankauszug (resource dump) aus den gefundenen Einträgen zu generieren kann man folgende Abfrage formulieren, die sich je nach verwendeten Datenmodell im Graph unterschiedlich formuliert. Hier werden leere Knoten wieder rausgefiltert und Autorennamen und Informationen abgefragt die eine ist- Teil- von- Beziehung aufweisen:

```
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX dcmitype: <http://purl.org/dc/dcmitype/>
PREFIX dcq: <http://purl.org/dc/qualifier/1.0/>
PREFIX gnd: <http://d-nb.info/gnd/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX fabio: <http://purl.org/spar/fabio/>
PREFIX urn: <http://www.d-nb.de/standards/urn/>
PREFIX shg: <http://localhost/view/>
```

```
## resource dump.
construct {
  ?p ?o <subject> .
  ?o ?x ?y .
} where {
  ?p ?o .
  optional {
    ?o ?x ?y
    FILTER (isBlank(?o) && !isBlank(?y))
  }
  optional {
    dct:isPartOf ?o .
    ?o ?x ?y .
  }
  optional {
    dct:creator ?o .
    ?o ?x ?y .
  }
}
```

Da Ausgaben von SPARQL- Abfragen im XML- Format möglich sind, ist der letzte Schritt über eine XSLT- Transformation zu erreichen. Das so generierte Ergebnis in dem Format solr.xml kann von Solr indexiert und damit über das Interface VuFind gefunden werden. Hatop hat seine Methode mit Erfolg an LOD- Sätzen der DNB getestet.

Die Sichtbarmachung von semantisch angereicherten Metadaten ist eine der besonderen Aufgaben moderner bibliothekarischer Tätigkeit. Allerdings ist SPARQL für die meisten Endnutzer schwer nachvollziehbar. Eine neuere Entwicklung ist das 2014 in erster Version erschienene JSON- LD, ein vom W3C als Standard anerkanntes Datenaustauschformat für Linked Data dass die Komplexität von RDF wesentlich reduziert, zu diesem aber kompatibel ist.¹⁴⁴

¹⁴⁴ Vgl.: W3.org, 2014, JSON-LD 1.0. [online]. 2014. [Abgerufen 31 Januar 2016]. Verfügbar unter: <https://www.w3.org/TR/json-ld/>

5.2 Beispiel II: Einbindung von Patentinformationen in ein institutionelles Repositorium

Die Sichtbarmachung und Vermarktung eigener Patentanmeldungen spielt gerade für amerikanische Universitäten eine bedeutende Rolle für das Image und haben ein ökonomisches Potential.¹⁴⁵ Bei dem folgenden Projekt sollte für die Fondren Bibliothek¹⁴⁶ der Rice- Universität, einer privaten Universität in Texas, die in diversen Rankings zu den besten der USA zählt, eine Möglichkeit gefunden werden die Patentinformationen aus der Datenbank der USPTO zu beziehen und in dem eigenen Repositorium abzubilden. Dabei sollten drei Schritte automatisiert ablaufen, nämlich die Identifizierung, der Download und schließlich der Upload der Universitäts- Patente. Alle Schritte sollten mit Open- Source- Software oder Web- Portalen möglich sein. Die bisherige Suche nach Rice- Patenten über Google's Patent- Search wurde als ungenügend empfunden. Die Datenbank des amerikanischen Patentamts bildet nur Patente bis 1980 ab und schließt fehlerhafte Metdatensets zu einzelnen Patenten aus, was laut Projektgruppe für eine nicht bestimmbare Menge von Rice- Patenten zutrifft. Als Vorteil der Abbildung in einem Forschungsrepositorium wäre außerdem zu nennen das der gesamte Forschungsprozess eines Anmelders über die Archivverlinkung sichtbar werden würde. Über einen langen Zeitraum hinweg können Zitationsanalysen mit den Repositoriumsdaten den Einfluss des jeweiligen Patents auf die damit zusammenhängende Forschung und Entwicklung abbilden.

In einer ersten Etappe musste evaluiert werden wie viele Patente im Zusammenhang mit Rice überhaupt existieren. Mit der webbasierten Suche über PubWEST des USPTO wurden 200 Patente gefunden über die Eingabe angenommener Anmelder- Informationen mit den Strings „Rice University“ oder „William Marsh Rice University“. Über die Schlagwort – Suche und die Ergänzung einer privaten vorgehaltenen Liste mit Patentanmeldungen die bis 1978 zurückreichen konnte die Anzahl der Patentmeldungen durch die Rice- Forschungsgemeinschaft auf 365 erhöht werden. Nächstes Problem waren die Anforderungen des lokalen Metadatenprofils (siehe Abb. 5) dass 16 Felder umfasst und aus einer Kombination von INID Codes¹⁴⁷ und

¹⁴⁵ Für diesen Aspekt und den folgenden Abschnitt vgl.: CARLSON, Scott; SPIRO, Linda. Collecting and Describing University-Generated Patents in an Institutional Repository: A Case Study from Rice University. *Code4Lib Journal*, 2015, Nr. 30.

¹⁴⁶ Vgl: Library.rice.edu, 2016, Welcome | Rice University | Fondren Library. [online]. 2016. [Abgerufen 31. Januar 2016]. Verfügbar unter: <http://library.rice.edu/>

¹⁴⁷ Inid- Codes werden zur Normierung von Patentliteratur eingesetzt. Diese sind Nummern zur Identifikation bibliographischer Daten nach dem WIPO- Standard ST.9, vgl.: Web2.wipo.int, 2016, Histo-

R = required field.
M = multiple values allowed.
A = automatically generated by repository software.

- i. [dc.contributor.assignee \(R, M\)](#)
- ii. [dc.contributor.publisher \(R\)](#)
- iii. [dc.creator \(R, M\)](#)
- iv. [dc.date.filed \(R\)](#)
- v. [dc.date.issued \(R\)](#)
- vi. [dc.description.abstract \(R\)](#)
- vii. [dc.digitization.specifications \(R\)](#)
- viii. [dc.format.extent \(R\)](#)
- ix. [dc.identifier.citation \(A\)](#)
- x. [dc.identifier.patentID \(R\)](#)
- xi. [dc.language.iso \(R\)](#)
- xii. [dc.type \(R\)](#)
- xiii. [dc.title \(R\)](#)
- xiv. [dc.type.dcmi \(R\)](#)
- xv. [dc.type.genre \(R\)](#)

Abbildung 4: Das Metadatenprofil des Patent- Repositoriums der Rice- Universität, Texas
Quelle: <https://digitalriceprojects.pbworks.com>

Dublin- Core besteht die natürlich nicht alle im angebotenen Metadatenexport von PubWEST angeboten werden können, was auch auf die Abstracts zutrifft. Mithilfe der Webseite Free Patents Online¹⁴⁸ die einen umfangreichen Metadatenexport aus den Ergebnisdokumenten einer Suche erlaubt und der Kombination der Metadatenexporte von PubWEST wurden die Metadaten über das Open- Source- Tool OpenRefine¹⁴⁹ zusammengeführt und normalisiert. Das nächste Problem war dass das USPTO Downloads nur in Bündeln erlaubt die alle innerhalb einer Woche angemeldeten Patente umfasst. Es wurde sich entschieden automatisiert PDF- Kopien aus der Volltext- und Bilder- Datenbank (PatFT¹⁵⁰) zu beziehen. Das USPTO bietet über eine HTML- Repräsentation der Volltexte den Download von eingebetteten PDF's. Bei der Betrachtung des Seitenquelltext fiel auf dass jede eingebettete vollständige PDF- Kopie die Endung „.0.pdf“ aufwies. Der Verzeichnispfad für jedes Patent besteht außerdem aus der

ry of WIPO Standard ST.9. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://web2.wipo.int/wipostad/en/standards/st9/history>

¹⁴⁸ Vgl. Freepatentsonline.com, 2016, FPO IP Research & Communities. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://www.freepatentsonline.com/>

¹⁴⁹ Vgl.: Openrefine.org, 2016, OpenRefine. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://openrefine.org/>

¹⁵⁰ Vgl.: Patft.uspto.gov, 2016, patft » Page 1 of 1. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://patft.uspto.gov/netahhtml/PTO/>

siebenstelligen, rückwärts notierten Patent- ID- Nummer und hat die folgende sich für jedes Patent wiederholende HTML- Syntax¹⁵¹:

```
<embed src="http://pimg-fpiw.uspto.gov/fdd/56/234/051/0.pdf" width="100%" height="850" type=application/pdf></embed>
```

Im nächsten Vorgang wurden aus den Patentbezeichnern URI's generiert. Dafür wurden die Verzeichnispfad- Abschnitte getrennt in einer Excel Tabelle aufgeführt und mit der Verkettens- Funktion zusammengeführt und so eine Liste aus PDF- URL's entstehen zu lassen.

A	B	C	D	E	F	G	H	I
US8880707B2	8880707	07	807	0	88	0.pdf	07/807 /088/0.pdf	http://pimg-fpiw.uspto.gov/fdd/07 /807/088/0.pdf

Abbildung 5: Tabelle mit Patent- ID und URL

Quelle: Carlson 2015

Mit dem Open- Source- Tool GNU Wget,¹⁵² einem reinen Konsolenprogramm wurden schließlich die Downloads realisiert, wobei eine Funktion des Programms die Umbenennung der Dateien von „0.pdf“ zu der jeweiligen Patent- ID erledigte. Der Tabelle wurde eine weitere Spalte hinzugefügt um einen Wget- Befehl zu generieren unter Verkettung von Spalte A1 (Patent ID) und Spalte I1 (erzeugte URL) mit:

=CONCATENATE(“wget -O “,A1,“.pdf“,I1)

Die so gewonnenen Wget- Befehle hatten folgerichtig die Syntax:

wget -O Patent-ID.pdf http://pimg-fpiw.uspto.gov/fdd/umgekehrtePatent-ID.0.pdf

Mit der Kopie der Wget- Spalte in eine Text- Datei die anschließend als ausführbare .bat- Datei gespeichert wurde, konnte der Prozess abgeschlossen werden. Alle Patente wurden automatisiert über die URL gefunden und beim Download umbenannt. Da die

¹⁵¹ Das hier gezeigte Beispiel wurde aus dem Quelltext der HTML- Seite für das Patent US005123456 entnommen, vgl.: Pdfpiw.uspto.gov, 2016, Patent Images. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://pdfpiw.uspto.gov/.piw?PageNum=0&docid=05123456&IDKey=8F265804A27C&HomeUrl=http%3A%2F%2Fpatft.uspto.gov%2Fmetahtml%2FFPTO%2Fpating.htm>

¹⁵² Vgl.: Gnu.org, 2016, GNU Wget. [online]. 2016. [Abgerufen 31 January 2016]. Verfügbar unter: <https://www.gnu.org/software/wget/>

so gewonnen Downloads nur als Image- Scans vorlagen mussten diese noch über die Recognize- Text Funktion von Adobe Acrobat händisch einem OCR- Prozess unterzogen werden da DSpace nur als Text erkannte Daten als Bit- Streams speichert.

Beide Beispiele zeigen deutlich welchen Zweck OSS in sehr speziellen Anwendungsbereichen erfüllen kann. Oft sind geschickte Verknüpfungen mehrerer Anwendungen erforderlich. Während Beispiel 1 fundierte Kenntnisse semantischer Technologien und sehr gute Kenntnisse in Java voraussetzt ist für Beispiel 2 eine Kombination aus Wissen über HTML, EXCEL und den Funktionalitäten der OSS- Tools Wget und OpenRefine notwendig.

6 Fazit / Ausblick

In dieser Arbeit wurde dargestellt, welche Problemfelder den Einsatz von OSS in Bibliotheken beeinflussen um eine fundierten Aussage über das Potential treffen zu können. Da es für bestimmte Aspekte wie der Marktanalyse zu Bibliothekssoftware aber an aussagekräftigen Daten mangelt, kann nur eine Tendenz beschrieben werden. Zur Verdeutlichung werden in Anlehnung an das Regulationsmodell von Lessig nochmals die vier Kräfte und deren festgestellte Ausprägungen auf OSS zusammengefasst.

1. Der Markt

Wie bereits angedeutet, fehlt eine ausreichende Datenbasis aus der man eine Potentialaussage ableiten könnte. Die Tendenz ist dass OSS in allen Branchen professioneller Informationsdienstleistung zunehmend mehr Anwendung erfährt, die auch über die Ebene der Infrastruktur hinaus geht. Dies gilt insbesondere für institutionelle Repositorien wissenschaftlicher Bibliotheken in den USA und Deutschland. Bei der Anwendung cloudbasierter BMS/ILS ist in beiden Ländern eine erneute Hinwendung zu proprietären Anbietern zu beobachten, es gibt aber wie das Quali OLE- Projekt gezeigt hat, auch hier Möglichkeiten die Anforderungen modernen Datenmanagements mit OSS erfolgreich zu erfüllen. Bei öffentlichen Bibliotheken in den USA gehört OSS mehr und mehr zum Alltag und der Marktanteil ist relativ hoch. Für den öffentlichen Sektor in Deutschland konnte man nur in Stadtverwaltungen eine breitere Anwendung erkennen, aber hier ist der Trend in den letzten 5 Jahren rückläufig. Für öffentliche Bibliotheken im besonderen lässt sich keine Aussage treffen.

2. Gesetze/Recht

In Deutschland erlaubt die gesetzliche Lage die vermehrte Anwendung von OSS problemlos und patentrechtliche Streitfälle sind nicht in größerem Ausmaß zu erwarten. Durch eine geschickte Auswahl vorhandener Open- Source-Lizenzen lassen sich trotz bisher unklarer Richtlinien im Vergaberecht durch das Urheberrecht abgesicherte Verhältnisse schaffen. Trotzdem muss auf erneute Änderungen der Regelungen zum geistigen Eigentum frühzeitig reagiert werden und evtl. neue Lizenzmodelle integriert werden. Die Strategien der europäischen Kommission, die Open- Source forcieren, können ein wesentlicher Orientierungspunkt sein. In Amerika ist durch die Patentrechtsproblematik der rechtliche Rahmen von FOSS tendenziell unstabil. Der starke und offensive politische Rückhalt seitens des weissen Hauses und anderer Institutionen lässt aber die Annahme zu, dass dieses Problem sich relativieren könnte.

3. Normen

Die gesamtgesellschaftliche Tendenz zu einer Kultur des Teilens (Shareconomy) und damit der Aufweichung des proprietären Modells kommt den durch die Open- Source- Gemeinschaft vertretenen sozialen Standards zugute. Die ethischen und technologischen Grundsätze der Bibliothekswelt und der Open-Source- Welt ergänzen sich.

4. Architektur/Code

Die zunehmende geforderte Interoperabilität aller Daten benötigt für ihre Verwirklichung offene Standards, Formate, Protokolle und Sprachen. Die neue Architektur des semantischen Netzes benötigt einen sichtbaren und veränderbaren Quellcode zur Realisierung. Bibliotheken in den USA und Deutschland haben hierzu mit Linked Open Data bereits einen wichtigen Beitrag geleistet müssen sich aber neue Informationstechnologien mehr zu eigen machen und eigenes Know- How entwickeln um zu bestehen. OSS kann dabei ein Vehikel sein.

Diese Arbeit wurde ausschließlich mit OSS erstellt.

7 Anhang

Anhang I: Transkript

Interview mit Jens Mittelbach, Abteilungsleiter Benutzung und Service der SLUB Dresden am 16.12.2015, 11:30 – 12:30 Uhr via Skype, Transkription von Volker-Simon Best

I: Was ist aus Ihrer Sicht als Bibliothekar der größte Vorteil von Open-Source-Software gegenüber proprietärer Software?

S: Der größte Vorteil liegt auf der Hand, die Openness. Man kann in den Quellcode reingucken, man kann den Quellcode analysieren. Man kann, selbst wenn man nicht [5] programmiert, was allerdings selten vorkommen dürfte, denn wenn man in der Lage ist Quellcode zu analysieren, ist man natürlich auch versucht den Quellcode so anzupassen, dass es den eigenen Bedürfnissen entspricht, das ist halt der größte Vorteil. Wir haben in der ... im bibliothekarischen Umfeld es eben sehr häufig traditionell mit proprietären Systemen, Softwaresystemen zu tun, und zwar nicht nur in [10] den Randbereichen unserer Tätigkeit sondern auch in den zentralen Bereichen. Als zentralen Bereich würde ich eben zum Beispiel das Datenmanagement, das Managen von Daten und Metadaten bezeichnen. Wenn wir hierfür nur proprietäre Systeme zur Verfügung haben, dann ist es sehr sehr problematisch weil wir ... und zwar aus zweierlei Hinsicht, weil wir einerseits natürlich gerne diese Software soweit [15] anpassen wollen dass sie mit unseren lokalen Bedürfnissen übereinstimmt, dass sozusagen unsere lokalen Bedürfnisse abbildet und unsere lokalen Funktionalitäten (.), die also umsetzen können. Das geht eben mit proprietärer Software insofern schlecht dass wir nicht in den Quellcode rein können und eben auch nicht dürfen. In solchen Fällen haben wir Reverse Engineering gemacht, haben versucht aus dem Verhalten der [20] Software erstmal Schlüsse zu ziehen welche Prozesse ablaufen, wie die Algorithmen genau aussehen könnten. Und dann wenn wir das Verhalten ändern wollen entsprechend Workarounds zu bauen, also Softwarehüllen ringsum nochmal um das Produkt zu legen, die sozusagen das Verhalten nachdem es schon generiert ist aus dem Softwarekern nochmal umbiegen, umändern oder umleiten so dass man am Ende [25] also ein halbwegs gewünschtes Verhalten hat. Das ist natürlich extrem problematisch, weil diese Software, weil nicht alle Eventualitäten bedacht werden. Man kann nicht genau sagen in welchen Fällen die Software sich genau wie verhält sondern, dass ist mehr so eine Trial and Error-Methode. Dann ist es aus einer anderen Sichtweise auch noch problematisch wenn dann die Software sich verändert, wenn es [30] ein Update gibt, wenn eine neue Version kommt, muss die Software die ringsum als Hülle liegt und die Workarounds herstellt eben auch aktualisiert werden, in den allermeisten Fällen. Das ist eben dann am Ende was die Total Costs of Ownership betrifft ein äußerst ungünstiges Verhältnis. Dann bezahle ich sozusagen für die Software selbst, also die im Bibliothekssektor üblichen recht hohen Lizenzgebühren [35] und dann bezahl ich die Entwicklerkapazitäten die im Prinzip das Verhalten der Software, weil es nicht befriedigend ist, eben noch mal verändern. Das sind laufende Kosten weil sich die Software ja auch entwickelt. Also das ist der erste Aspekt warum Software, warum Openness für bibliothekarische Anwendungen sinnvoll ist. Also die

Möglichkeit diesen Quellcode selbst zu beeinflussen und selbst daran mitzuschreiben. [40]Der zweite Aspekt ist, wenn ich Software selbst mit entwickeln kann, wenn ich in die Software reingucken kann, wenn ich also ergründen kann was in der Software vor sich geht weil ich die Algorithmen sehe die programmiert worden sind, dann wächst natürlich auf meiner Seite auch Know-How. Dieses Know-How ist dann genau das was wir brauchen insbesondere in den Kernbereichen der bibliothekarischen Tätigkeit. [45]I: Es gibt ja seit kurzem die Ausbildung zum Systembibliothekar. Man könnte sich also hinstellen und sagen es gibt ein paar Systembibliothekare die machen das, die kümmern sich um diese Problem und die normalen Bibliothekare arbeiten weiter wie bisher.

S: Unser Umfeld ist, also das Feld der Informationsprovider, nenne ich jetzt mal [50]Bibliotheken so, ist ... wir gehören ja zur Informationswirtschaft in einer gewissen Weise, ist derartig technologiegetrieben und bestimmt dass wir eigentlich diesen Ansatz nicht mehr verfolgen können. Wir können nicht sagen, wir beschäftigen so ein paar Programmierer, also Hanseln die sich halt mit der Technik auskennen und die eigentlichen Bibliothekare die, was weiß ich, die machen was anderes, die gehen halt [55]mit den Büchern um oder wie auch immer. Das hat früher funktioniert, dass funktioniert aber jetzt nicht mehr weil im Prinzip die ganze Landschaft digital ist. Nicht nur im Bereich des Datenmanagements also in den klassischen Feldern des Verzeichnens und Erschließens sondern auch in den Bereichen des Vermittelns hält die Digitalisierung Einzug, digitalisiert sich unsere Landschaft. Es wird mittlerweile mehr [60]oder weniger offen und unverhohlen von Digital Scholarship gesprochen. Selbst die Geisteswissenschaften kennen das inzwischen, haben es zumindest schon mal gehört auch wenn sie noch viel Angst davor haben. Die Digitalisierung hält auf allen Ebenen unserer Gesellschaft unserer Welt derart mächtig Einzug, dass wir es uns nicht leisten können als Informationsanbieter nicht auch ganz substantielles Know-How [65]aufzubauen. Das geht eigentliche nur wenn wir die Technologien verstehen. Wenn wir die technischen Konzepte auch verstehen, gerade im Datenmanagement also ich meine Linked Open Data als Konzept, semantische Technologien als Konzept, das ist etwas was wir Bibliothekare also ganz zuinnerst, zutiefst, verstehen müssen um solche Konzepte nutzbar zu machen und im Prinzip könnte es so sein, dass die [70]bibliothekarischen Ressourcen und die bibliothekarischen Angebote das Fundament der Wissenswelt darstellen. Man muss sich auch mal vergegenwärtigen dass Bibliotheken das Wissen von Jahrtausenden beherbergen und sich diese Welt des Internets und was dort entsteht, ständig neu entsteht, und da entsteht ganz viel dass man natürlich auch sagen. Der Informationszuwachs ist da natürlich enorm, es werden [80]ja in zwei Jahren mehr Daten erzeugt als in den gesamten zweitausend Jahren vorher (.) , aber trotzdem das Fundament ist eben dass was in Jahrtausenden gewachsen ist in den Bibliotheken beherbergt ist und das müssen wir so aufschließen und erschließen dass die Welt da draußen das wirklich auch als ein sicheres Wissensfundament benutzen kann. Sonst bauen die nämlich wieder auf Sand oder neu, [90]oder auf Fundamente die eigentlich schon da sind von denen sie aber gar nichts wissen. Das ist also sozusagen unsere Aufgabe und da in diesem Linked-Data-Bereich spezielle Konzepte zu entwickeln, das wäre eine der ureigenen Aufgaben von Bibliothekaren aber dazu müssen wir das erst mal verstehen, das Konzept. Verstehen tun wir das nur aus einer sehr technischen Sicht also man muss sich bis auf [100]Programmierersicht damit auseinandersetzen. Und das können wir mit proprietärer Software schlechter weil wir so das Gefühl vermittelt bekommen 'Was Du hast ein Problem? Hier ist die Lösung. Kauf Dir die Lösung.' Wenn man dann die Lösung hat und sagt naja, das Problem ist erst halb gelöst 'Na dann kauf dir noch ne andere Lösung, dann hast du das Problem gelöst.' Du bist nie in der Lage das Problem [105]vollkommen zu verstehen und die Lösung dafür selbst zu konzipieren.

I: Das erinnert mich an die Aussagen Richard Stallmanns der ja auch gesagt hat, dass Open-Source-Software mehr ist als nur ein technologisches Konzept, sondern dass es auch eine Philosophie, eine Politik dahinter gibt. Was haben Sie dazu zu sagen? Und da sie Linked-Data angesprochen haben, wo sehen sie noch Potential in Bezug auf das [110]Semantic Web?

S: Die Philosophie der Offenheit ist ein ganz wichtiger Punkt. Also ich glaube dass Bibliotheken (.) die stehen ja unter Existenzdruck unter Legitimationsdruck also die müssen sozusagen noch beweisen das sie noch eine Funktion haben können in dieser modernen Welt. Viele Wissenschaftler, gerade in den Naturwissenschaften, nehmen [115]Bibliotheken ja gar nicht mehr wahr, weil in deren Infrastrukturen Bibliotheken als Position gar nicht mehr vorkommen. Die sagen wir schreiben unsere Studien und veröffentlichen die dann, was weiß ich, auf arxiv.org und ich finde da auch alles was ich brauche um zu forschen. In die Bibliothek gehe ich gar nicht mehr. Die Bibliothek ist ein Ort wo Bücher stehen und Bücher brauche ich nicht. Bücher sind viel zu [120]langsam sozusagen. Insofern herrscht ein Bild von Bibliothek dass wir zum Großteil noch bedienen und dass den Keim der Zerstörung einer Legitimation ganz tief in sich trägt. Jetzt kommt die Offenheit mit ins Spiel. Also wenn wir als Bibliotheken, als Anbieter von Informationen als Verteidiger der Offenheit von Information, des Zugangs von Information, der Offenheit von Ressourcen, der Offenheit von Wissen, [125]der Offenheit auch von Prozessen, wissenschaftlichen Prozessen, als Vertreter von Open-Access, Open-Data und Open-Science auftreten als Förderer einer Wissenschaftskultur der Offenheit, dann ist darin ein ganz tiefer Daseinszweck den wir erfüllen können. Der auch rechtfertigt dass Bibliotheken öffentlich gefördert werden und politische eine Rolle spielen. Also Bibliotheken als Garanten für eine [130]Demokratisierung der Gesellschaft weil eben dass was an Information zugänglich ist allen zur Verfügung steht und alle das für demokratische Prozesse auch notwendige (.), die notwendigen Kompetenzen aufbauen können. Insofern ist Open-Source-Software und die Beförderung von Open-Source-Software ein Teil der Offenheits-Philosophie die wir ganz ganz wichtig finden. Insofern hat OSS einen ganz [135]praktischen Nutzen in dem wir gucken und verändern was die Software macht aber auch sozusagen eine philosophische Ebene als Teil der Openness-Bewegung. Bibliotheken als Garant von Offenheit sind quasi schon verpflichtet dazu, zumindest in den Kern-Tätigkeitsbereichen Open-Source einzusetzen und mit zu entwickeln und damit auch zu fördern. Das wäre also diese philosophische Komponente. Felix [140]Lohmeier und ich haben dazu auch kürzlich einen Aufsatz geschrieben Offenheit statt Bündniszwang. Noch mal zum zweiten Aspekt Semantic Web und Linked-Data. Ja, es ist wirklich so dass die Bibliotheken da schon was geliefert haben und im Prinzip die Bibliotheken auch diejenigen sind die den Impuls gesetzt haben oder setzen in Bezug auf diese Technologie. Wenn ich das recht beobachte ist es wohl auch so, [145]dass das draußen in der Welt der Webentwickler weniger bekannt ist als im Bibliothekswesen. Was das Ganze natürlich auch schwieriger macht, weil die Webentwickler diejenigen sind die die Entwicklung treiben, also die dieses Ganze, also diesen Flickenteppich von Leuten die irgendwas entwickeln und sich quasi in der Zusammenarbeit und in ihrem Metier Quasi-Standards herausgebildet haben, [150]Interoperabilitätsvorstellungen herausgebildet haben, dass in deren Gesichtskreis Linked-Open-Data noch nicht in dem Maße angekommen ist wie wir uns das wünschen. Dass die sagen 'was ihr bietet Linked-Data an? Ihr habt ne SPARQL-Schnittstelle? Wir können damit was anfangen, wir können eure Schnittstelle für unsere Zwecke benutzen.' Dass dies noch nicht der Fall ist, beobachten wir schon. Wir [155]müssen uns selber überlegen wie machen wir das noch nutzbarer dass diese Daten die wir erzeugen und pflegen in der Welt der Webentwickler nutzbar werden. Insofern sind Bibliotheken Vorreiter im Bereich Linked-Data aber sie sind es noch

nicht genug. Es reicht eben nicht Katalogdaten die wir jetzt haben, relativ gut strukturiert, es reicht eben nicht diese Daten als Linked-Data über eine Schnittstelle [160]verfügbar zu machen oder runterladbar zu machen, weil eben damit noch keiner wirklich was anfangen kann. Keine Endnutzer, sowieso nicht, aber auch die Zwischennutzer die die Daten benutzen könnten um irgendwelche Anwendungen zu bauen, um tolle Webservices zu erzeugen. Man denkt da zum Beispiel an die Tourismusbranche die mit historischen Reiseführen die wir digitalisiert haben und den [165]Daten die da drin sind, also auch den Volltexten, wunderbare Sachen machen könnten, wunderbare Apps machen könnten, die Leuten ermöglichen wenn sie durch eine Stadt gehen über ihre Handy-Fotolinse, also was gerade in das Blickfeld kommt eine Wissenswelt zu offerieren. Das könnte man alles auf der Grundlage dieser Daten machen. Das heißt, wir müssen den Grad der Semantisierung verstärken, wir müssen [170]diese Verknüpfungen, das sind ja im Prinzip zwei Dinge. Das eine ist die Technologie, unsere Daten als Linked-Data zur Verfügung zu stellen, das andere ist die Qualität, also Semantizität, ist damit noch nicht garantiert. Also Semantizität ist erst da wenn die Daten weiter hochverknüpft sind. Wenn die Beziehung zwischen den Objekten die in den Informationseinheiten bestehen, also die Frage ist über wie viele [175]Verknüpfungen ... und im Prinzip ist das ja nur ein Modell der Wirklichkeit. In Wirklichkeit also, also die Gewebemetapher ist eigentlich falsch. In Wahrheit ist das so eine, wie soll ich das nennen, ist die Frage, wie Wissen miteinander verknüpft ist, ist eben nicht über einzelne Maschen, einzelne Kanten, sondern über ein ganz dichtes Geflecht, fast wie so eine Art Suppe wenn Sie so wollen. Das modellhafte Abbild [180]dessen ist ja Linked-Open-Data, die LOD-Cloud, dabei sind die wirklich gezogenen Verknüpfungen zwischen diesen einzelnen Wissensseinheiten, Informationseinheiten, und das ist natürlich viel lückenhafter, maschenartiger und gewebeartiger bestenfalls, sind fast wie so ein Spinnennetz mit ganz vielen Lücken durch die man auch durchrutschen kann. Diese Lücken zu verkleinern bis zu einer [185]gewebeartigen, suppenartigen Konsistenz die die Wirklichkeit so gut wie möglich abbilden das wäre eine Aufgabe. Semantic-Technologies sind noch überhaupt nicht an ihrem Ende sondern erst am Anfang würde ich sagen. Allerdings nur wenn wir auch was dafür tun, das es eben weitergeht. Sonst werden Webentwickler da draußen sagen ‘nein das ist ein falsches Versprechen da kommt ja nichts, das ist zu dünn und zufällig [190]diese Verknüpfungen die da immer als Semantic bezeichnet werden, wir machen andere Technologien, wir machen das was wirklich greifbar ist.’ Die sind ja viel pragmatischer als Bibliothekare. Die müssen ja irgendwie Resultate haben, während wir solchen Idealzuständen nachstreben die letztlich ja auch nicht zu erreichen sind. Wir müssen also Treiber dieser Entwicklung sein. Das heißt zum einen die [200]Datenqualität zu verbessern, die Anzahl der semantischen Verknüpfungen zwischen den Informationseinheiten zu erhöhen, und dann zweitens den Zugriff darauf auch besser zu gestalten. Also nicht über SPARQL sondern über entsprechende Webtechnologien, also JSON und solche Geschichte die einfach da draußen auch verstanden werden. Und da gibt es Ansätze, also es gibt Ansätze Linked-Data in JSON [205]so zu verpacken, dass man letztlich draußen in der Welt der Webentwickler was damit anfangen kann.

I: Sind wir als Bibliothek also bereits daran gescheitert?

S: Ich würde das noch nicht als gescheitert betrachten, aber wir können dafür sorgen dass es scheitert wenn wir so weiter machen wie bisher. Also ständig das Wort im [210]Munde rumführen, indem wir unsere Kataloge als RDF und Linked-Data zur Verfügung stellen aber nicht dafür sorgen, dass es benutzbar wird und selbst nicht auch dafür sorgen dass wir Anwendungen darauf setzen. Da kommen wir auch wieder zur Open-Source-Software. Die traditionellen Systeme sind überhaupt nicht willens mit Linked-Data wirklich umzugehen, weil die sich verkaufen wie geschnitten Brot. Weil

[215]die Gewinnmargen so groß sind dass man keine Veranlassung sieht diesen Bereich weiter zu entwickeln, gegenwärtig jedenfalls.

I: Aber der Knowledge Graph von Google geht doch in diese Richtung, oder?

S: Ja genau, bei Google geht man in diese Richtung. Die machen Linked-Data, die sind an der Semantisierung von Daten interessiert weil sie eben sehen dass man damit [220]letztlich genauer das Informationsbedürfnis bedienen kann. Natürlich ist die Maschine im Vorteil die versteht was der Nutzer fragt gegenüber der Maschine die einfach nur String-Matches ausgibt und nicht versteht was diese Strings bedeuten oder wie sie mit anderen Bedeutungseinheiten in Verbindung stehen. Wir müssen ja auch mal überlegen dass wir auf dem Weg sind zu einer künstlichen Intelligenz, also der [225]Anspruch ist ja schon dass die Suchmaschine auch mit meiner natürlich-sprachigen Anfrage umgehen kann, dass ich der sagen kann gib mir alle Autoren die in der Zeit von 1920-1934 zur Problematik der Pressetätigkeit geschrieben haben. Das sind so Sachen da wollen wir ja hin dass die Maschine das versteht und nicht nur ein einfaches String-Match daraus macht. Google und solche Firmen wie Google gehen in [230]diese Richtung die das aber immer pragmatisch sehen, die machen das nicht aus einer rein philosophischen Sicht sondern die gucken vom Nutzen und was sich für eine Anwendung daraus wird. Die bieten dann eben so etwas an wie ihren Knowledge-Graph, was sich für den Nutzer ja nur darin sichtbar macht, dass rechts in einer Box die Verknüpfungen des Suchbegriffs sichtbar werden. Das müssen wir eben für uns [235]auch so annehmen, dass es nicht nur darum geht unten drunter die Infrastrukturen so zu machen, dass das theoretisch geht, sondern es muss auch praktisch gehen. Es muss durchschlagen an der Suchoberfläche von Bibliothekskatalogen und da, wie gesagt, sind wir bei der Open-Source-Software. Da müssen wir auch selbst entwickeln können und sagen können jetzt haben wir die Daten alle so und jetzt können wir das [240]und das damit machen. Damit das auch geht benutzen wir diese Open-Source-Software und entwickeln die so dass eben so geht wie wir das wollen. Dann kommen wir vielleicht auch weg von solchen Ansätzen weg wie 'Wir bilden das Wissen der Welt graphisch ab.' Also der Begriff Graph der jetzt noch nichts mit der visuellen Darstellung ..., also das bedeutet ja nicht dass man das jetzt visuell darstellen muss. [245]Sondern der bedeutet erst mal nur dass die Dinge in netzartigen Strukturen miteinander in Verbindung stehen und damit kann man nun alles Mögliche machen. Zum Beispiel kann man das so machen wie Google das macht mit dieser Infobox die da rechts steht. Man kann es auch anders machen. Man kann datengetriebene Abfragen machen und sagen also der Nutzer kann jetzt so einen Follow-your-Nose-Approach [250]machen. Der kann sagen ich suche jetzt nach einem Begriff und dann finde ich in dem Katalog das und das, also eine Liste von Dingen und aus diesen gefundenen Daten und den Daten die wiederum damit in Verbindung stehen ergibt sich ein Weg und der wird mir gezeigt. Da wird so eine Facette eingeblendet die mir dynamisch aus dem gefundenen Datenbestand generiert wird und wo gesagt wird mit Deiner Abfrage, [255]mit den Treffern die sich daraus ergeben, ergeben sich diese und jene Wege, die kannst Du abgehen.

I: Sie meinen also einen anderen Ansatz als die momentan noch häufig genutzten Filter Bubbles?

S: Ja genau das wäre wichtig, dass man in Zukunft Systeme zum Explorieren anbietet [260]und nicht zum Filtern.

I: Meinen Sie auch hier haben die Bibliotheken einen Auftrag?

S: Ich würde ganz klar sagen ja, die haben da auf jeden Fall einen Auftrag. Aber die Frage ist ob sie in der Lage sind den wahrzunehmen, den auszufüllen oder auszuführen. Was wieder zu Open-Source führt. Das können wir nur über Know-How [265]und Know-How können wir aufbauen wenn wir sagen dass wir uns nicht auf Fremdanbieter verlassen die uns etwas halbfertiges anbieten das wir nehmen müssen

weil es eben nicht anders geht. Das wir das selber machen. Das wir eben auch durch das Selbermachen lernen wie es geht. Das ist ja auch ein Prozess. Weil wir wissen es ja vorher nicht. Das ist auch die Erfahrung die wir in Dresden machen mit den Open-
[270]Source-Softwaresystemen die wir entwickeln und mitentwickeln.

I: Die SLUB Dresden wurde bei uns in Seminaren öfter genannt wenn es um die Anwendung von PDA ging. Was setzten sie wirklich an Open-Source-Software ein?

S: Im Bereich PDA setzen wir momentan keine OSS ein, sollten das aber tun und wollen das in Zukunft auch, weil das ebenso ein Kernbereich ist, Bestandsaufbau.
[275]Oder die Frage wie letztlich die Ressourcen die wir anbieten, also wo die herkommen oder wer uns sagt wo es die gibt oder wer bestimmt ob wir sie anbieten, das ist also so ein Kernbereich wo wir uns also nicht auf Fremdanbieter verlassen sollten die uns geschlossene Systeme vor die Füße setzen. Es gibt ja auch die Möglichkeit eine Firma zu beauftragen Software zu entwickeln die dann aber nicht
[280]geschlossen ist, das machen wir zum Teil auch. Ein Beispiel dazu wäre unser 3D-Rauminformationssystem das wir einsetzen. Wenn Sie in unseren Katalog reingehen, dann finden Sie zu jedem Freihandbestand, zu jedem Freihandtitel- oder Exemplar so eine kleine Karte wo das Buch sich im Bibliotheksraum befindet und wenn Sie da drauf klicken und können in so ein navigables 3D-System eintauchen wo sie
[285]tatsächlich diesen Raum explorieren, also virtuell.

I: Also so eine Art Life 2.0- Anwendung quasi?

S: In gewisser Weise schon. Zumindest kann man es dahin entwickeln. Augmented-Reality-Anwendung. Auf jeden Fall ist das auch ein Weg den physischen Raum Bibliothek im digitalen Raum abzubilden und zugänglich zu machen. Und das machen
[290]wir mit einer Firma, Makongo, zusammen aber nach Grundsätzen die letztlich auch diesem Offenheitsprinzip folgen. Dann gibt es auch Sachen die wir regelrecht selbst oder mitentwickeln, zum Beispiel GOBI unsere Digitalisierungs-Workflow-Software für die Digitalisierung unserer Objekte, alten Drucke und Handschriften und so weiter. Außerdem D:Swarm, das ist eine Datenmanagementlösung die dazu da ist
[295]letztendlich das bibliothekarische Datenmanagement zu gestalten. Das sind auch so Sachen wie SLUB-Semantics wo wir mit einer Firma zusammen, aber auf der Grundlage von OSS und die semantische Anreicherungen von Katalogdaten gewährleisten.

I. Ist das jetzt auch eine Management-Richtlinie, dass sie sagen wir machen
[300]Kollaborationen mit Unternehmen aus der Privatwirtschaft, aber nur wenn diese sich dem Open-Source-Gedanken verpflichtet fühlen?

S: Das machen wir zunehmend, ja. Weil wir gelernt haben mit unserem Discovery-System, da haben wir ja ein proprietäres System lizenziert, gekauft und eingesetzt und haben eben daran gelernt das wir vieles von dem was wir wollen damit nicht machen
[305]können. Viele Versprechungen die die Verkäufer dieser Software machen, die Anbieter dieser Software machen, sich letztlich einlösen und das wir zum Teil sehr spezifischen Anforderungen haben, die damit nicht abdeckbar sind und wir letztlich mehr investieren als wir bei einer Open-Source-Lösung investieren. Bei diesem Kernbereich sind wir also dahin gekommen eine Lösung anzuwenden bei der Open-
[310]Source drüber steht. Auch in Kooperation mit kommerziellen Anbietern. Das ist ja nicht per se schlecht. Wir haben ja nichts gegen die Wirtschaft, das Missverständnis soll bitte nicht aufkommen, dass wir sagen die Leute die für Produkte Geld nehmen sind per se übel. Das meinen wir gar nicht. Die Ressourcen die wir in unserem Verantwortungsbereich haben, die aus der öffentlichen Hand gefördert werden, und
[315]damit schon bezahlt sind, müssen... dass die auch frei bleiben. Wer eine gute Geschäftsidee hat und eine schöne App baut und Daten von uns verwendet um damit Mehrwertdienste zu gewährleisten der soll das bitte gerne verkaufen, da haben wir

überhaupt nichts dagegen. Was wir nicht wollen ist dass das was uns eigentlich gehört uns nicht noch mal hinterher noch mal verkauft wird, mehrfach sogar.

[320]I: Also nicht so wie das bei vielen Verlagsmodellen ist, bei denen man für seine eigenen Daten doppelt zahlt?

S: Wenn wir uns mal als gemeinschaftlichen Wissenschaftskörper verstehen, als Körper zu dem wir als Hostler und Anbieter, als Archivare der Wissenschaft gehören, und produzieren, es verschenken um es dann wieder kaufen zu müssen damit es [325]zugreifbar wird, das soll nicht sein. Das können wir in vielen Fällen nur dann machen wenn wir auf solche Open-Source-Lösungen setzen weil wir dann Herr unserer Daten sind. Weil wir dann bestimmen was damit geschieht. Wir sind da durchaus liberal. Wir sollten eben dann nicht den Fehler machen den kommerzielle Anbieter machen und sagen weil wir vom Staat wenig gefördert werden, weil die [330]Personalressourcen und die Budgets immer weiter zurückgeschnitten werden verkaufen wir eben das was jetzt schon eigentlich der Öffentlichkeit gehört. Das darf eben auch nicht sein. Sondern wir müssen uns die Legitimationen erarbeiten die uns zum Verfechter des öffentlichen Eigentums machen. Was der Öffentlichkeit gehört soll auch der Öffentlichkeit gehören und das ist unsere Aufgabe denke ich. Und was man [335]dann alles damit machen, mit diesem Eigentum, wie man das weiter verwerten kann, das ist eine andere Frage. Da kommt dann die Wirtschaft mit ins Spiel.

I: Was haben Sie für Erfahrungen mit den Kosten für OSS? Es wird ja unter anderem gesagt der erhöhte Konfigurationsaufwand mache die fehlenden Lizenzkosten nicht wett.

[340]S: Man muss es immer vom Gesichtspunkt des Return of Investment sehen oder den Total Cost of Ownership. Diese Frage muss man betriebswirtschaftlich-empirisch beantworten, man kann nicht so aus dem Bauch heraus sagen das kostet mehr oder dies kostet weniger. Man muss genau gucken was kostet was, also auf den gesamten Prozess bezogen. Oft erscheint, oft ist der Blick auf die Total Cost of Ownership [345]verstellt vor den Entscheidungsträgern. Also die sehen ich zahle hier eine Million für eine Archivierungslösung die eine Langzeitarchivierung erlaubt und wenn ich das alles selbst entwickeln müsste brauche ich 10 Jahre und 7 Stellen, dann kostet mich das am Ende 2 Millionen. Also kauf ich die Lösung. Dabei hat man dann aber vernachlässigt dass die Lösung, die man gekauft hat für eine Million gar nicht im Haus [350]ist und einem auch nicht insofern gehört, dass sich dann die Betreuung erledigt hätte. Ich muss ja dann immer noch reininvestieren, in Updates investieren. Ich muss in die Leute investieren, mit der Firma kommunizieren und sagen ‘ Jetzt passt mal auf wir brauchen dass so und so‘ oder die Firma setzt dass was wir wollen nicht um also müssen wir es selbst machen um die Workarounds zu haben damit die Software etwas [355]macht was funktionell erst mal gar nicht vorgesehen ist und dann kommt man am Ende schnell bei drei Millionen an. Dann ist klar dass die Investition von zwei Millionen die von drei Millionen sticht. Und das man dann empirisch betrachten... Muss also wirklich genau da Daten erheben, sich dann auch fragen welchen Grad von Flexibilität möchte ich eigentlich? Ist es weil Langzeitarchivierung mein Kernbereich [360]ist, sinnvoll mich in die Arme eines kommerziellen Anbieters zu begeben dessen erste Intention ist damit Geld zu verdienen und der mir nicht zusagen kann ob er mir langfristig, im Sinne der Langzeitarchivierung zur Verfügung steht? Gibt es die Firma morgen noch? Oder in einem Jahr? Hat die das Projekt abgestoßen weil die sehen damit ist kein Blumentopf zu gewinnen? Das sind die Fragen die man sich [365]beantworten muss. Da kommt man eben in der Praxis sinnvollerweise bei einer Mischlösung an. Es gibt bestimmte Dinge die sind nicht prioritär und nicht Kernbereich und deshalb kaufen wir uns da Lösungen zu ein, die sind auch nicht lebenskritisch oder überlebenskritisch. Wenn ich jetzt also dieses 3D-Modell nehme und ich kann das (.) Es ist eben nicht so dass die Bibliothek nicht mehr funktioniert

[370]weil dieses 3D-Modell nicht gepflegt wird. Die Bibliothek hört auf zu existieren wenn es keinen ordentlichen Katalog mehr gibt, der den Zugriff und das Entdecken auf die Objekte ermöglicht. Deshalb ist der Katalog ein Kernbereich und deshalb setzen wir da auch auf eine Open-Source-Software, sind dabei auf eine Open-Source-Lösung umzustellen. Das sind die Fragen. Natürlich ist das Kostenargument wichtig, aber es [375]ist keinesfalls so einfach zu beantworten, so nach dem Motto wenn ich das outsource dann wird das alles viel billiger. Openness heißt nicht Kostenfreiheit, das muss man auch mal sagen. Ein System der Wissenschaftskommunikation das auf Openness-Gedanken basiert heißt nicht, dass das nichts kostet. Im Gegenteil das kostet auch was. Weil es Leute gibt die bezahlt werden müssen, die dieses System am Leben [380]erhalten und, aber der Vorteil von Open-Source ist eben noch mal, dass ich selbst bestimme was ich mache und wie ich es mache und was ich zu jedem gegebenen Zeitpunkt die Sache kosten lasse während ein geschlossenes System das Problem der Systemzwänge hat. Ich kann eben bei geschlossenen Systemen nicht einfach sagen das ist mir jetzt zu teuer, ich will die Kosten reduzieren. Da stehe ich dann meistens vor [385]der Wahl das ganz abzuschalten, das ganze System, kann also unter Umständen einen Kernbereich nicht mehr bewirtschaften. Das können wir uns als Bibliotheken eben nicht leisten. Wir müssen das was wir als Bibliotheken machen müssen auch absichern. Das geht mit Open-Source-Ansätzen unserer Überzeugung nach besser, sicherer oder auch nachhaltiger.

[390]I: Ich frage mich manchmal, also wenn man sich etwa den Open-Access-Server SCIELO in Mittel- und Südamerika ansieht mit 1,5 Millionen Downloads täglich, wie schnell und erfolgreich da der Open-Access-Gedanke auch mit Hilfe von Open-Source umgesetzt wurde (.) Ist das vielleicht auch ein Haltungs- oder Motivationsproblem?

S: Allerdings. Das ist ein Haltungs-, Motivations- und Verständnisproblem. All das [395]was ich gesagt habe trifft nur zu wenn wir uns gemeinschaftlich dem verschreiben und gemeinschaftlich erkennen dass da eine Chance drin liegt. Solange es so ist wie es im Moment ist, dass wir sagen 'Also wisst ihr die Wissenschaftler zwingen uns ja dass wir solche Konsortialverträge abschließen, solche lizenzierten Ressourcen lizenzieren' weil die das brauchen, und wir zahlen jeden Preis weil die [400]Wissenschaftler das brauchen, und wir schließen uns in im Konsortium zusammen und zahlen immer noch jeden Preis weil die Wissenschaftler das brauchen... und diejenigen die ein OA-Repository aufbauen, den Open-Access-Gedanken verfolgen, die sind zwar bewunderungswürdig aber in der Minderheit und das kann das nicht ersetzen was wir bislang machen. Solange es also noch genügend [405]Entscheidungsträger gibt die das so sehen wird das auch so sein. Also wir können die Wende auch nur herbeiführen, wenn wir uns diesen Prinzipien ehrlich verschreiben und das befördern. Insofern ist es ein Motivations- aber auch Verständnisproblem weil es eben viele... , weil viele kurzfristig sagen so eine Kauflösung kostet mich jetzt weniger, das kann ich mir jetzt leisten mit dem Budget was ich jetzt habe und so einen [410]Open-Source-Lösung ist aufwendiger und kostet mehr, solange ich also nur diesen Implementierungsprozess betrachte. Es kostet ja sozusagen auch erst mal mehr weil ich das Know-How das dafür notwendig wäre noch nicht im Haus habe, weil ich ja langfristig auch in den Personalaufbau investieren müsste. Ich müsste halt auch sagen ich stelle jetzt Leute ein die das können. Ich stelle jetzt also keinen [415]Diplombibliothekar ein der super katalogisieren kann sondern einen Software-Entwickler der in der Lage ist ein System zu programmieren dass automatisch katalogisiert, dass automatisch Fremddaten aus dem Web abgreift und ich halte so ein Buch nur noch vor die Kameralinse und dann habe ich gleich die Metadaten dazu. Und das machen wir ja nicht. Wir stellen weiter Diplombibliothekare ein die das Buch nicht [420]vor die Linse halten sondern sich hinsetzen und das abtippen was im Buch steht, oder wie das auch ist, da ist jetzt verflacht ausgedrückt aber im Prinzip ist das so.

I: Es gibt ja auch das Problem der Curricula. Der neue Studienjahrgang lernt erst jetzt RDA usw. Wie sehen sie das?

S: Das sind systemische Prozesse, da müsste man systemisch anfangen. Eben bei der [425]Ausbildung anfangen. Das bedeutet das die Bibliotheksdirektoren, die Leute in den Ministerien die müssen das verstehen, deshalb ist das so ein Verständnisproblem. Die müssen das verstehen und wollen und nicht nur Lippenbekenntnisse dazu leisten.

Anhang II

Purpose of System	Respondents using a system for this purpose	Respondents using an OSS solution	Percent respondents using an OSS solution
Institutional repository	69	52	75.4%
Blogging	65	51	78.5%
Digital preservation	57	50	87.7%
Publishing	57	42	73.7%
Authentication/identity management	67	35	52.2%
Digital asset management	64	34	53.1%
Web analytics	71	22	31.0%
Discovery layer	73	19	26.0%
Study room scheduler	59	18	30.5%
ELMS	23	17	73.9%
Streaming media	62	17	27.4%
Data warehouse	26	14	53.8%
Visualization	27	10	37.0%
Electronic resource management	71	9	12.7%
Link resolver	70	9	12.9%
Floor maps	43	8	18.6%
Data analysis	32	7	21.9%
ILS	74	6	8.1%
Course reserve	68	4	5.9%
Inter-library loan	73	3	4.1%

Abbildung 6: Benutzte OSS der befragten ARL- Bibliotheken nach Anwendungsbereich

Quelle: Thacker 2015

8 Literaturverzeichnis

- ALLISON RANDAL, 2015, The Future of Open Source. [online]. 2015. [Abgerufen 23 Januar 2016]. Verfügbar unter: <http://allisonrandal.com/2015/07/22/the-future-of-open-source/>
- ALTENHÖNER, Reinhard. Deutschland und die USA: Aspekte der Informationsinfrastruktur im Wechselspiel von Orientierung und Abgrenzung. *Bibliothek Forschung und Praxis*, 2015, 39. Jg., Nr. 2, S. 150-157.
- ALTENHÖNER, Reinhard. Jenseits der Cloud: Metadaten- und Datenmanagement in der bibliothekarischen Infrastruktur Teil 1. *Bibliotheksdienst*, 2015, 49. Jg., Nr. 7, S. 677-695.
- AMERICAN LIBRARIES MAGAZINE, 2015, Library Systems Report | American Libraries Magazine. [online]. 2015. [Abgerufen 13 Januar 2016]. Verfügbar unter: <http://american-librariesmagazine.org/2015/05/01/library-systems-report/>
- AYRE, Lori Bowen. MARC isn't Dying Fast Enough. *Collaborative Librarianship*, 2015, 6. Jg., Nr. 4, S. 175-177.
- AZURE.MICROSOFT.COM, 2016, Microsoft showcases the Azure Cloud Switch (ACS) | Microsoft Azure Blog. [online]. 2016. [Abgerufen 26 Januar 2016]. Verfügbar unter: <https://azure.microsoft.com/de-de/blog/microsoft-showcases-the-azure-cloud-switch-ac/>
- B-I-TONLINE.de, 2016, b.i.t.online: Bibliotheksfachzeitschrift mit Information und Technologie. [online]. 2016. [Abgerufen 12 Januar 2016]. Verfügbar unter: <http://www.b-i-t-online.de/daten/nuernberg/KN-2015-interview-lubeek>
- BERGMANN, Julia; DANOWSKI, Patrick (Hg.). *Handbuch Bibliothek 2.0*. Walter de Gruyter, 2010.
- BLISCHKE, Johannes. *Betriebssysteme für Smartphones: Android vs. iOS*. Bachelor+ Master Publication, 2015.
- BOYLE, James. A Politics of Intellectual Property: Environmentalism for the Net?. *Duke Law Journal*, 1997, 47. Jg., Nr. 1, S. 87-116.
- BREEDING, Marshall, 2016, Library Technology Guides: Documents, Databases, News, and Commentary. *Librarytechnology.org* [online]. 2016. [Abgerufen 13 Januar 2016]. Verfügbar unter: <http://librarytechnology.org>
- BREEDING, Marshall, 2016, Library Technology Guides: Documents, Databases, News, and Commentary. *Librarytechnology.org* [online]. 2016. [Abgerufen 16 Januar 2016]. Verfügbar unter: <http://librarytechnology.org>
- BREEDING, Marshall. The open source ILS: still only a distant possibility. *Information Technology and Libraries*, 2002, 21. Jg., Nr. 1, S. 16.
- BSZ-BW.DE, 2016, Bibliothekssysteme im Bibliotheksservice-Zentrum Baden-Württemberg: Koha. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <https://www.bsz-bw.de/bibliothekssysteme/koha.html>
- BUECHEREI-PRAXISHANDBUCH.DE, 2016, Bücherei Praxishandbuch: Bibliothekssoftware. [online]. 2016. [Abgerufen 27 Januar 2016]. Verfügbar unter: <http://www.buecherei-praxishandbuch.de/index.php?id=51>
- BUXMANN, Peter; DIEFENBACH, Heiner; HESS, Thomas. *Die Softwareindustrie: Ökonomische Prinzipien, Strategien, Perspektiven*. Springer-Verlag, 2015.
- C.EUROPA.EU, 2016, Urheberrecht in der Informationsgesellschaft - Europäische Kommission. [online]. 2016. [Abgerufen 20 January 2016]. Verfügbar unter: http://ec.europa.eu/internal_market/copyright/copyright-infso/index_de.htm#maincontent-Sec1
- CARLSON, Scott; SPIRO, Linda. Collecting and Describing University-Generated Patents in an Institutional Repository: A Case Study from Rice University. *Code4Lib Journal*, 2015, Nr. 30.
- CASSELL, Mark. Why governments innovate: Adoption and implementation of open source software by four European cities. *International public management Journal*, 2008, 11. Jg., Nr. 2, S. 193-213.
- CODE4LIB, 2011, About. [online]. 2011. [Abgerufen 30 Januar 2016]. Verfügbar unter: <http://code4lib.org/about>

- CONFERENCES.OREILLY.COM 2016, O'Reilly Open Source Convention. [online]. 2016. [Abgerufen 29 Januar 2016]. Verfügbar unter: <http://conferences.oreilly.com/oscon>
- CONRAD- REMPEL, Steffi, 2015, Opus 4 in Zahlen und Fakten beim KOBV. In : 104. *Deutscher Bibliothekartag* [online]. Nürnberg. 2015. [Abgerufen 19 Januar 2016]. Verfügbar unter: <https://opus4.kobv.de/opus4-bib-info/frontdoor/index/index/docId/1936>
- DATAHUB.IO, 2016, Search for a Dataset - the Datahub. [online]. 2016. [Abgerufen 30 Januar 2016]. Verfügbar unter: <https://datahub.io/dataset?tags=lod>
- DE.SLIDESHARE.NET, 2016, 2015 Future of Open Source Survey Results. [online]. 2016. [Abgerufen 11 Januar 2016]. Available from: http://de.slideshare.net/North_Bridge/2015-future-of-open-source-study
- DFG.DE, 2016, DFG - Deutsche Forschungsgemeinschaft - Formulare und Merkblätter / Gesamtliste. [online]. 2016. [Abgerufen 27 Januar 2016]. Verfügbar unter: http://www.dfg.de/foerderung/formulare_merkblaetter/index.jsp
- DIGITAL-LAW-ONLINE.INFO, 2016, CONTU Final Report. [online]. 2016. [Abgerufen 20 Januar 2016]. Verfügbar unter: <http://digital-law-online.info/CONTU/PDF/index.html>
- DIPBT.BUNDESTAG.DE, 2016, DIP21 Extrakt. [online]. 2016. [Abgerufen 28 Januar 2016]. Verfügbar unter: <http://dipbt.bundestag.de/extrakt/ba/WP18/653/65399.html>
- DWHEELER.COM, 2016, Why Open Source Software / Free Software (OSS/FS, FOSS, or FLOSS)? Look at the Numbers!. [online]. 2016. [Abgerufen 28 January 2016]. Verfügbar unter: http://www.dwheeler.com/oss_fs_why.html
- EC.EUROPA.EU, 2016, Open Source Strategy in the European Commission: History - DIGIT. [online]. 2016. [Abgerufen 27 Januar 2016]. Verfügbar unter: http://ec.europa.eu/dgs/informatics/oss_tech/history/history_en.htm
- ENGARD, Nicole. Practical open source software for libraries. Elsevier, 2010.
- ENIS, Matt. Ending the invisible Library. *Library Journal*, 2015, 140. Jg., Nr. 3, S. 36-38.
- ETTRICH, Matthias. Koordination und Kommunikation in Open-Source-Projekten. *Open Source Jahrbuch*, 2004, S. 179-192.
- FREEPATENTSONLINE.COM, 2016, FPO IP Research & Communities. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://www.freepatentsonline.com/>
- FUNK, Stefan E.; NEUBAUER, Mathias. Formatcharakterisierung. In: NEUROTH, Heike. nestor Handbuch. *Boizenburg: Verlag Werner Hülsbusch*, 2009.
- GESETZE-IM-INTERNET.DE, 2016, UrhG - Einzelnorm. [online]. 2016. [Accessed 20 January 2016]. Available from: https://www.gesetze-im-internet.de/urhg/_2.html
- GITHUB, 2016, The White House. [online]. 2016. [Abgerufen 28 Januar 2016]. Verfügbar unter: <https://github.com/whitehouse>
- GNU.ORG, 2016, GNU Wget. [online]. 2016. [Abgerufen 31 January 2016]. Verfügbar unter: <https://www.gnu.org/software/wget/>
- GNU.ORG, 2016, What is free software?- GNU Project - Free Software Foundation. [online]. 2016. [Abgerufen 8 Januar 2016]. Verfügbar unter: <http://www.gnu.org/philosophy/free-sw.html>
- GRASSMUCK, Volker; SIBER, Karl Heinz. *Freie Software: Zwischen Privat-und Gemeineigentum*. Bundeszentrale für politische Bildung, 2004.
- HATOP, Götz. Integrating Linked Data into Discovery. *Code4Lib journal*, 2013, 21. Jg., S. 67-90.
- HBZ-NRW.DE, 2016, hbz — Das Bibliotheksmanagementsystem Quali OLE – Überblick und aktueller Stand. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: https://www.hbz-nrw.de/dokumentencenter/veroeffentlichungen/bms_kuali_ole_2015.pdf/view
- HBZ-NRW.DE.de, 2016, hbz — Entwicklungspartnerschaft zwischen hbz, VZG und der Quali OLE Community. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: https://www.hbz-nrw.de/aktuelles/entwicklungspartner_ole
- HEISE ONLINE, 2014, Studie zur Softwarequalität: *Open Source schlägt proprietär*. [online]. 2014. [Abgerufen am 16 November 2015]. Verfügbar unter: <http://www.heise.de/newsticker/meldung/Studie-zur-Softwarequalitaet-Open-Source-schlaegt-proprietaeer-2175954.html>
- HILLENIIUS, GIJS, Markus Feilner, 2016, Europas Behörden vertrauen zunehmend auf Open-S... - Seite 5 » Linux-Magazin. *Linux-Magazin* [online]. 2016. [Abgerufen 28 Janu-

- ar 2016]. Verfügbar: <http://www.linux-magazin.de/Ausgaben/2013/11/Open-Source-in-Europa/%28offset%29/8>
- HTC Global Services. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <http://www.htcinc.com/home.asp>
- INFORMATIK UND GESELLSCHAFT, TU Berlin, 2016, Open Source Jahrbuch. [Opensourcejahrbuch.de](http://www.opensourcejahrbuch.de) [online]. 2016. [Abgerufen 13 Januar 2016]. Verfügbar unter: <http://www.opensourcejahrbuch.de/project/index.html>
- JAEGER, Till; SCHULZ, Carsten. Gutachten zu ausgewählten rechtlichen Aspekten der Open Source Software. *Webseite: http://www.ifross.de/ifross_html/art47.Pdf (05.01. 2016)*, 2005.
- JAFFE, Lee David; CAREAGA, Greg. Standing up for open source. *Library Philosophy and Practice*, 2007, 9. Jg., Nr. 2, S. 21.
- JENA.APACHE.ORG, 2016, Apache Jena -. [online]. 2016. [Abgerufen 30 Januar 2016]. Verfügbar unter: <http://jena.apache.org/>
- JOINUP.EC.EUROPA.EU, 2016, 'Freiburg's frustration with IT caused anger over free software' | Joinup. [online]. 2016. [Abgerufen 27 Januar 2016]. Verfügbar unter: <https://joinup.ec.europa.eu/news/freiburgs-frustration-it-caused-anger-over-free-software>
- JOINUP.EC.EUROPA.EU, 2016, Vatican library: open source for long-term preservation | Joinup. [online]. 2016. [Abgerufen 27 January 2016]. Verfügbar unter: <https://joinup.ec.europa.eu/community/osor/news/vatican-library-open-source-long-term-preservation>
- KEES, Alexandra. *Open Source Enterprise Software*. Springer Fachmedien Wiesbaden, 2015.
- KOHA-COMMUNITY.ORG, 2016, Official Website of Koha Library Software. [online]. 2016. [Abgerufen 26 Januar 2016]. Verfügbar unter: <https://koha-community.org/>
- KROES, Neelie. Being open about standards. *OpenForum Europe. Brussel. Verfügbar unter: <http://europa.eu/rapid/pressReleasesAction.Do>*, 2008.
- KUALI FOUNDATION, info @ kuali.org, 2016, Functionality by Module. *Kuali.org* [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <http://www.kuali.org/ole/modules>
- KUALI FOUNDATION, info @ kuali.org, 2016, Kuali Enterprise Workflow (KEW). *Kuali.org* [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <https://www.kuali.org/rice/modules/kew>
- KUHLEN, Rainer. Wie öffentlich soll Wissen für Wissenschaft und Unterricht sein?. *Anmerkungen zum Urheberrecht in der Informationsgesellschaft*. In: EIBL, Maximilian. WOLFF, Christian (Hrsg.); WOMSER-HACKER, Christa (Hrsg.): *Designing Information Systems: Festschrift für Jürgen Krause*. Konstanz: UVK Verlagsgesellschaft mbH, 2005, 221. Jg., S.27-45.
- LEHMANN, Frauke. FLOSS developers as a social formation. *First Monday*, 2004, 9. Jg., Nr. 11.
- LESSIG, Lawrence. *Code and other laws of cyberspace*. Basic Books, New York 1999.
- LESSIG, Lawrence. *Code: And Other Laws of Cyberspace, Version 2.0*. 2006.
- LIBRARY.RICE.EDU, 2016, Welcome | Rice University | Fondren Library. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://library.rice.edu/>
- LINUXCOUNTER.NET, 2016, The Linux Counter Project - Statistics about Linux, its Users and more. [online]. 2016. [Abgerufen 9 Januar 2016]. Verfügbar unter: <https://www.linuxcounter.net/>
- LINUXFOUNDATION.ORG, 2016, Linux Trademark Institute. [online]. 2016. [Abgerufen 25 Januar 2016]. Verfügbar unter: <http://www.linuxfoundation.org/programs/legal/trademark>
- LOHMEIER, Felix; MITTELBACH, Jens, 2014, Offenheit statt Bündniszwang. *Zeitschrift für Bibliothekswesen und Bibliographie*. 2014. Jg. 61, Nr. 4-5, S. 209-214.
- LUTTERBECK, Bernd. Die Zukunft der Wissensgesellschaft. *Lutterbeck, Bernd et al., Open Source Jahrbuch*, 2006, S. 445-465.
- MAGGS, Peter B. The Uncertain Legal Status of Free and Open Source Software in the United States. In: *Free and Open Source Software (FOSS) and other Alternative License Models*. Springer International Publishing, 2016. S. 477-493.
- MERGES, Robert P., *Software & Internet Law*. Aspen Publishers, 2011.

- MILLINGTON, Peter, 2016, OpenDOAR - Home Page - Directory of Open Access Repositories. *OpenDOAR.org* [online]. 2016. [Abgerufen 18 Januar 2016]. Verfügbar unter: <http://www.opendoar.org/>
- MITTELBACH, Jens, 2011, Zur Zukunft von Bibliothekssoftware. *Nbn-resolving.de* [online]. 2011. [Abgerufen 12 Januar 2016]. Verfügbar unter: <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-66156>
- MITTELBACH, Jens. *Modernes Datenmanagement: Linked Open Data und die offene Bibliothek*. o-bib. Das offene Bibliotheksjournal/herausgegeben vom VDB, 2015, 2. Jg., Nr. 2, S. 61-73.
- MORGAN, Eric Lease. Possibilities for open source software in libraries. *Information Technology and Libraries*, 2002, 21. Jg., Nr. 1, S. 12.
- NETMARKETSHARE.COM, 2016, Operating system market share. [online]. 2016. [Abgerufen 11 Januar 2016]. Verfügbar unter: <http://www.netmarketshare.com/operating-system-market-share.aspx>
- NEWS.NETCRAFT.COM, 2016, Web Server Survey | Netcraft. [online]. 2016. [Abgerufen 22 Januar 2016]. Verfügbar unter: <http://news.netcraft.com/archives/category/web-server-survey/>
- NEWS.NETCRAFT.COM, 2016, Web Server Survey | Netcraft. [online]. 2016. [Abgerufen 9 Januar 2016]. Verfügbar unter: <http://news.netcraft.com/archives/category/web-server-survey/>
- NGINX, 2015, The Evolution of Open Source: Interview with NGINX CEO Gus Robertson. [online]. 2015. [Abgerufen 22 January 2016]. Verfügbar unter: <https://www.nginx.com/blog/the-natural-evolution-of-open-source-an-interview-with-nginx-ceo-gus-robertson/>
- O'REILLY, Tim, 2016, The Architecture of Participation - O'Reilly Media. *Archive.oreilly.com* [online]. 2016. [Abgerufen 30 Januar 2016]. Verfügbar unter: http://archive.oreilly.com/pub/a/oreilly/tim/articles/architecture_of_participation.html
- OPENREFINE.ORG, 2016, OpenRefine. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://openrefine.org/>
- OPENSOURCE.COM, 2016, US government accelerating development and release of open source. [online]. 2016. [Abgerufen 29 Januar 2016]. Verfügbar unter: <https://opensource.com/government/14/4/interview-david-wheeler>
- OPENSOURCE.ORG, 2016, Licenses & Standards | Open Source Initiative. [online]. 2016. [Abgerufen 22 Januar 2016]. Verfügbar unter: <http://opensource.org/licenses>
- OPENSOURCE.ORG, 2016, The Open Source Definition | Open Source Initiative. [online]. 2016. [Abgerufen 9 Januar 2016]. Verfügbar unter: <http://opensource.org/docs/osd>
- OSS-WATCH.AC.UK, 2013, Koha: a case study in open source project ownership. [online]. 2013. [Abgerufen 25 Januar 2016]. Verfügbar unter: <http://oss-watch.ac.uk/resources/cs-koha>
- PDFPIW.USPTO.GOV, 2016, Patent Images. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://pdfpiw.uspto.gov/piw?PageNum=0&docid=05123456&IDKey=8F265804A27C&HomeUrl=http%3A%2F%2Fpatft.uspto.gov%2Fnetathtml%2FPTO%2Fpatimg.htm>
- PEUKERT, Alexander; KÖNIG, Dominik. License Contracts, Free Software and Creative Commons in Germany. In: *Free and Open Source Software (FOSS) and other Alternative License Models*. Springer International Publishing, 2016. S. 201-226.
- PICOT, Henriette. Die deutsche Rechtsprechung zur GNU General Public License. *Lutterbeck, Bernd; Bärwolff, Mathias; Gehring, RA (Hg.): Open Source Jahrbuch*, 2008, S. 233-244.
- PUBLICADMINISTRATION.UN.ORG, 2016, Methodology. [online]. 2016. [Abgerufen 28 Januar 2016]. Verfügbar unter: <https://publicadministration.un.org/egovkb/en-us/About/Methodology>
- RAYMOND, Eric. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 1999, 12. Jg., Nr. 3, S. 23-49.
- RAZUM, Matthias, et al. eSciDoc infrastructure: a fedora-based e-Research framework. In: *Research and Advanced Technology for Digital Libraries*. Springer Berlin Heidelberg, 2009. S. 227-238.

- REGISTRY.DURASPACE.ORG, 2016, DSpace User Registry | DuraSpace. [online]. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <http://registry.duraspace.org/registry/dspace/>
- REGISTRY.DURASPACE.ORG, 2016, Fedora User Registry | DuraSpace. [online]. 2016. [Abgerufen 18 January 2016]. Verfügbar unter: <http://registry.duraspace.org/registry/fedora>
- REIMANN, Katlen. Aktuelle Marktübersicht des Bibliothekssoftwaremarktes in Deutschland unter Einbeziehung von Open Source Produkten. *Bachelorarbeit, Stuttgart. Online verfügbar unter: http://opus.bszbw.de/hdms/volltexte/2008/652/pdf/Bachelorarbeit.pdf (letzter Zugriff: 21.01. 2016), 2008.*
- SCHRAPE, Jan-Felix. *Open Source Softwareprojekte zwischen Passion und Kalkül.* Stuttgarter Beiträge zur Organisations-und Innovationsforschung, SOI Discussion Paper, 2015.
- SCHWEITZER, Roswitha and Kemnek-Heek, Kirstin, 2016, Das Open Source Bibliothekssystem Quali OLE : Ergebnisse der gemeinsamen Evaluation durch GBV und hbz. In : *104. Deutscher Bibliothekartag in Nürnberg* [online]. Nürnberg : Berufsverband Information Bibliothek. 2016. [Abgerufen 17 Januar 2016]. Verfügbar unter: <https://opus4.kobv.de/opus4-bib-info/frontdoor/index/index/docId/2078>
- SEAMAN, Graham. Adapting VuFind as a Front-end to a Commercial Discovery System. *Ariadne*, 2012, Nr. 68.
- SINGH, Vandana. Why migrate to an open source ILS? Librarians with adoption experience share their reasons and experiences. *Libri*, 2013, 63. Jg., Nr. 3, S. 206-219.
- SOURCEFORGE.NET, 2016, cvs free download - SourceForge. [online]. 2016. [Abgerufen 24 January 2016]. Verfügbar unter: <http://sourceforge.net/directory/os:linux/?q=cvs>
- SOURCEFORGE.NET, 2016, Enterprise Software. [online]. 2016. [Abgerufen 21 Januar 2016]. Verfügbar unter: <http://sourceforge.net/directory/enterprise>
- THACKER, Curtis; KNUTSON, Charles. Barriers to Initiation of Open Source Software Projects in Libraries. *Code4Lib Journal*, 2015, Nr. 29.
- THE LINUX MINT BLOG, 2016, Monthly News - January 2016. [online]. 2016. [Abgerufen 29 Januar 2016]. Verfügbar unter: <http://blog.linuxmint.com/?p=2985>
- THOMAS, Stephen W.; HASSAN, Ahmed E.; BLOSTEIN, Dorothea. Mining Unstructured Software Repositories. In: *Evolving Software Systems.* Springer Berlin Heidelberg, 2014. S. 139-162, S. 141.
- TORVALDS, STALLMANN:, 2015, Stallman: If you want freedom don't follow Linus Torvalds. *Computerworld* [online]. 2015. [Abgerufen 22 Dezember 2015]. Verfügbar unter: http://www.computerworld.com.au/article/195096/stallman_want_freedom_don_t_follow_linus_torvalds/
- UN, E. Government Survey. 2014. *E-government for the Future We Want-http://unpan3.un.org/egovkb/Portals/egovkb/Documents/un/2014-Survey/E-Gov_Complete_Survey-2014.pdf, Abgerufen januar, 2016, 28. Jg.*
- US, JOIN, 18F Digital Services Delivery. *18F Digital Services Delivery* [online]. 2016. [Abgerufen 28 Januar 2016]. Verfügbar unter: <https://18f.gsa.gov/>
- USPTO.ORG, 2016, United States Patent and Trademark Office. [online]. 2016. [Abgerufen 24 Januar 2016]. Verfügbar unter: <http://www.uspto.gov/>
- VAUGHAN-NICHOLS, Steven, 2014, Biggest patent win ever? Microsoft's billion dollar a year Samsung deal | ZDNet. *ZDNet* [online]. 2014. [Abgerufen 25 Januar 2016]. Verfügbar unter: <http://www.zdnet.com/article/biggest-patent-win-ever-microsofts-billion-dollar-a-year-samsung-deal/>
- VIGNOLI, Italo, 2016, LibreOffice: the numbers. *The Document Foundation Blog* [online]. 2016. [Abgerufen 9 Januar 2016]. Verfügbar unter: <https://blog.documentfoundation.org/blog/2016/01/01/libreoffice-the-numbers/>
- VUFIND.ORG, 2016, community:installations [VuFind Documentation]. [online]. 2016. [Abgerufen 19 Januar 2016]. Verfügbar unter: <https://vufind.org/wiki/community:installations>
- W3.ORG, 2014, JSON-LD 1.0. [online]. 2014. [Abgerufen 31 Januar 2016]. Verfügbar unter: <https://www.w3.org/TR/json-ld/>
- WEB.ARCHIVE.ORG, 2016, Computer Notes Magazine volume 1 issue 9 at the START-UP Gallery. [online]. 2016. [Aufgerufen 10 Januar 2016]. Verfügbar unter: <https://web.archi->

- ve.org/web/2012_032_3162213/http://startup.nmnaturalhistory.org/gallery/notesViewer.php?ii=76_2&p=3
- WEB2.WIPO.INT, 2016, History of WIPO Standard ST.9. [online]. 2016. [Abgerufen 31 Januar 2016]. Verfügbar unter: <http://web2.wipo.int/wipostad/en/standards/st9/history>
- WHITEHOUSE.GOV, 2015, Developers. [online]. 2015. [Abgerufen 28 Januar 2016]. Verfügbar unter: <https://www.whitehouse.gov/developers>
- WIKI.DOCUMENTFOUNDATION.ORG, 2016, Feature Comparison: LibreOffice - Microsoft Office - The Document Foundation Wiki. [online]. 2016. [Abgerufen 10 Januar 2016]. Verfügbar unter: https://wiki.documentfoundation.org/Feature_Comparison:_LibreOffice_-_Microsoft_Office
- WIKI.DURASPACE.ORG, 2016, Dashboard - DuraSpace Wiki. [online]. 2016. [Abgerufen 18 Januar 2016]. Verfügbar unter: <https://wiki.duraspace.org/dashboard.action>
- WINTERHALTER, Christian. Die EDV und ihre Systeme: Software. In: ALBERS, Christoph. *Erfolgreiches Management von Bibliotheken und Informationseinrichtungen: Fachratgeber für Bibliotheksleiter und Bibliothekare*. Verlag Dashöfer, 2013.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Berlin, den 01.02.2015

A handwritten signature in black ink, consisting of several overlapping loops and lines, positioned above a horizontal line.
