

Open Source Software in Bibliotheken – von Produkten zu flexiblen Infrastrukturen (Teil 1)

Matthias Finck

Einleitung

Open Source Software ist ein wichtiger Baustein im Portfolio bibliothekarischer Softwarelösungen – wie z.B. die Studie von Philipp Maaß anschaulich nachweist¹. Viele auch speziell für Bibliotheken zugeschnittene Systeme wie Kitodo, Opus, VuFind, WordPress oder TYPO3 haben sich am Markt etabliert. Allen gemein ist, dass diese Softwaresysteme in der Regel den spezifischen Bedarf genau einer bibliothekarischen Aufgabe abdecken. So ist Opus eine Speziallösung für Dokumentenserver, Kitodo unterstützt Digitalisierungsworkflows und VuFind bietet die Benutzungsschnittstelle für moderne Katalog-/Discoverysysteme.

Auch Open Source Software, die nicht für rein bibliothekarische Aufgaben entwickelt wurde, wird zumeist für bestimmte Aufgabenbereiche genutzt: WordPress zur direkten Kommunikation und TYPO3 zur Gestaltung der Website. Eine Ursache für die Systemvielfalt liegt darin, dass wann immer eine bibliothekarische Aufgabe durch bestehende Lösungen nicht abgedeckt werden konnte, zumeist ein neues System für die Unterstützung dieser Aufgabe implementiert wurde. Eine weitere Ursache liegt darin, dass sich in der Entstehungszeit der meisten einzelnen Spezialsysteme um die Jahrtausendwende herum kein System bereits als stabiles und etabliertes Werkzeug – und somit als Basis – durchgesetzt hatte.

Open Source Software aus der Produktperspektive

Diese aufgaben- bzw. produktorientierte Sicht auf die Systeme hat Landschaften entstehen lassen, in denen die Lösungen vertikal weitestgehend getrennt voneinander existieren – wie die Zusammenstellung verschiedener Open Source Systeme am Beispiel der Staats- und Universitätsbibliothek Hamburg (SUB) zeigt (vgl. Abb. 1).

¹ vgl. Maaß, Philipp (2016): Free/Libre/Open-Source Software in wissenschaftlichen Bibliotheken in Deutschland. Eine explorative Studie in Form einer Triangulation qualitativer und quantitativer Methoden. Masterthesis, Technische Hochschule Köln, <http://hdl.handle.net/10760/29324> (zuletzt besucht am 12.03.2018)

Bei der Nutzung von Open Source Software in Bibliotheken ist seit einiger Zeit der Trend zu erkennen, dass sich die Softwarelandschaften von der Produktorientierung zur Infrastrukturorientierung entwickeln. Dieser Veränderungsprozess von einzelnen Produkten hin zu flexiblen Infrastrukturen wird systematisch hinsichtlich seiner Vor- und Nachteile dargestellt und an zwei konkreten Beispielen (TYPO3/Fedora bzw. Folio/VuFind) deutlich gemacht.

Analyzing the use of open source software in libraries, the trend can be shown that the software landscape is evolving from standalone products to flexible infrastructures. This change process is systematically presented in this article and the advantages and disadvantages are discussed. Additionally, two concrete examples (TYPO3/Fedora or Folio/VuFind) point up the analysis.

Jedes System verfügt deshalb auch über den kompletten technischen Funktionsumfang: über eine eigene Datenhaltungsschicht, eine Verwaltungs- bzw. Redaktionsschicht und eine eigene Benutzungsschnittstelle. Über die Jahre entsteht so eine Vielfalt von Open Source Systemen, die alle für sich alleine stehen und alle einer entsprechenden Betreuung bedürfen. Diese Vielzahl getrennter Produkte birgt eine Reihe von Risiken bzw. Problemen – vor allem in der Nutzung und Betreuung:

Hoher Bedarf an Spezialisten: Sowohl in der Anwendung als auch in der Betreuung der Systeme braucht es eine hohe Spezialexpertise – mindestens eine Expertin oder einen Experten pro System, und zwar auf

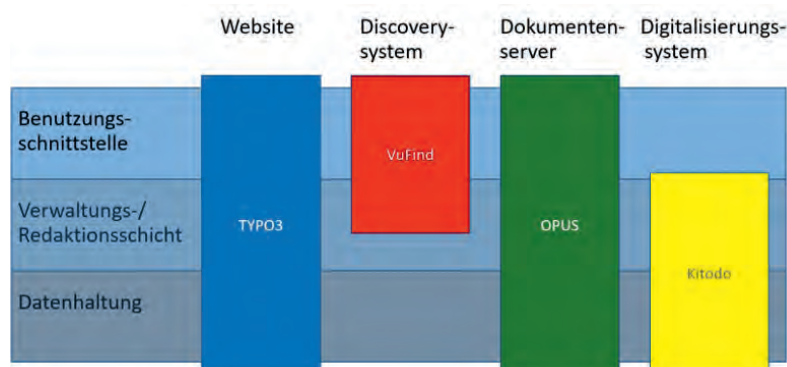


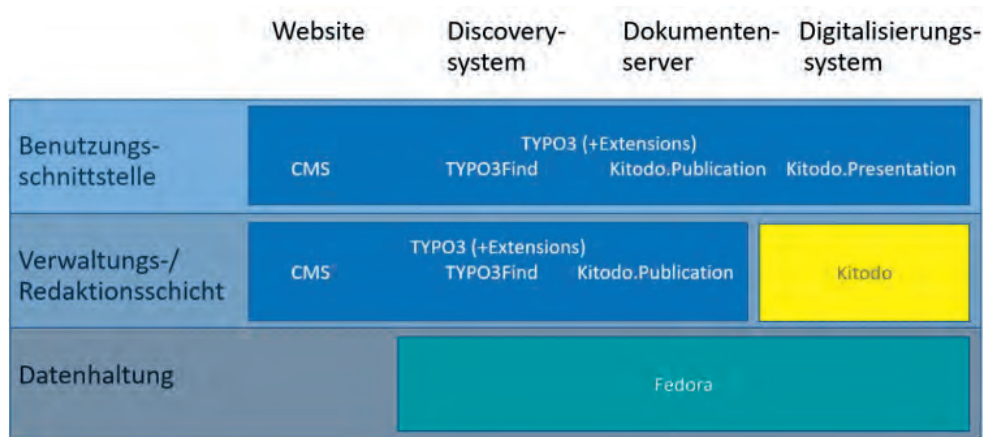
Abbildung 1: Exemplarischer produktorientierter Einsatz von Open Source Systemen der SUB

allen genannten Ebenen (Darstellung, Verwaltung, Datenhaltung). Die Systeme können z. B. in unterschiedlichen Programmiersprachen geschrieben sein oder andere Anforderungen an die Serverarchitektur stellen.

Heterogene Benutzungsschnittstellen: Da jedes System über eine eigene Benutzungsschnittstelle verfügt, ist es fast unmöglich den Nutzenden ein einheitliches Layout bzw. Design anzubieten. Zum einen verfügen die Systeme meist über unterschiedliche Template-Engines für die Oberfläche, mit unterschiedlichen Interaktionsmustern oder Sprachkonventionen. Zum anderen gelingt es nur unter hohem Aufwand, ein ein-

todo.Publication) oder Discoverysystemen (die Extension TYPO3Find) an der Benutzungsschnittstelle übernimmt. Gleichzeitig werden z.B. die dezentralen Datenhaltungen aufgehoben und mittels generischer Technologien wie Fedora oder Solr/ElasticSearch zusammgeführt. Aus einer Produktlandschaft mit Produkten, in der jedes für sich von der Datenhaltung bis zur Benutzungsschnittstelle alle Aufgaben übernimmt, werden Infrastrukturlösungen, die die Aufgaben horizontal teilen (Benutzungsschnittstelle, Verwaltungs-/Redaktionsschicht und Datenhaltung).

Diese Entwicklung von einzelnen Produkten hin zu flexiblen Infrastrukturen lässt sich an zwei Beispielen verdeutlichen (siehe Abb. 2).



Beispiel 1: Bibliotheksspezifische TYPO3-Extensions

Im TYPO3-Umfeld zeigt sich, dass immer mehr bibliotheksspezifische Erweiterungen entstehen, die die Benutzungsschnittstellen der früheren Systeme ersetzen. So übernimmt die Extension Kitodo. Publication weitestgehend die Funktionalität des redaktionellen Backends eines Dokumentenservers, während wahlweise Kitodo.Presentation

oder TYPO3Find in der Darstellung der Objekte zum Einsatz kommen. Große Teile der Benutzungsschnittstelle bzw. der Verwaltungs-/Redaktionsschicht werden von TYPO3 abgedeckt. Dabei nimmt TYPO3 weniger die Funktion eines Produkts für eine konkrete bibliothekarische Aufgabe ein, sondern stellt vielmehr einen Rahmen bzw. eine Infrastruktur bereit, in die sich diverse Extensions einklinken können. Der Vorteil liegt darin, dass alle Systeme nun auf dieselbe Template-Engine zur einheitlichen Darstellung zurückgreifen können und dass alle Subsysteme auf derselben technischen Basis laufen. Es bedarf so viel weniger technisch differenzierter Expertise.

Gleichzeitig wird die Datenhaltung von den beiden Interaktionsschichten getrennt und eine gemeinsame Datenhaltungsschicht eingeführt. Das heißt, dass alle erstellten Objekte, seien sie über einen Dokumentenserver hochgeladen oder im Rahmen eines Digitalisierungsprozesses entstanden, in einem gemeinsamen Datenbestand gesammelt werden und dadurch auch in rein fachlich motivierten Sammlungen präsentiert werden können. Ein Pfad, den die Sächsische Landesbibliothek – Staats- und Universitätsbibliothek Dres-

Abbildung 2: Prototypische TYPO3-/Fedora-basierte Open Source Infrastruktur

heitliches Design für alle Einzelsysteme zu entwickeln und dann aktuell zu halten.

Unvermittelbare Zugangsbeschränkungen: Die Systeme übernehmen teilweise überlappende Aufgaben. So werden den Nutzenden beispielsweise „born digital“ und „retrodigitalisierte“ Objekte zu einem Thema oftmals in zwei verschiedenen Sammlungen angeboten, weil die einen im Dokumentenserversystem erfasst wurden und die anderen über die Digitalisierungssoftware. Diese unterschiedlichen Zugangswege sind den Anwenderinnen und Anwendern kaum zu vermitteln.

Open Source Software als flexible Infrastruktur

Die Nachteile dieser produktorientierten Sichtweise sind lange bekannt und es gibt immer wieder einzelne Entwicklungen, die konsolidierende Lösungsansätze aufzeigen. Seit einiger Zeit verstärkt sich dabei der Trend, dass einzelne Softwaresysteme durch bibliotheksspezifische Extensions ihre Aufgabengebiete verbreitern, indem z.B. TYPO3 mittlerweile auch die Aufgabe von Dokumentenservern (die Extension Ki-

den mit dem „Medienübergreifenden Repository“² seit geraumer Zeit beschreitet.

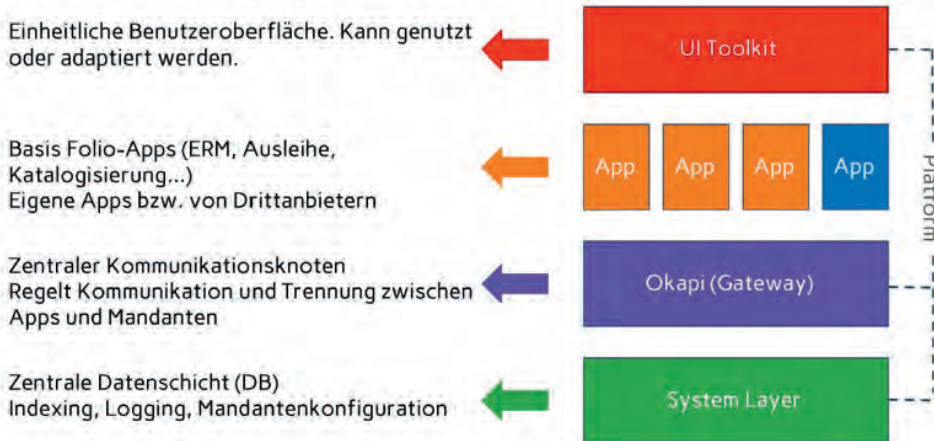
**Beispiel 2:
Das Foliokzept im
Zusammenspiel mit VuFind**

Eine ganz ähnliche Entwicklung lässt sich in dem Zusammenspiel von Folio und VuFind beobachten. In diesem Fall übernimmt Folio die Rolle des Infrastruktursystems, welches statt Extensions in diesem Fall sogenannte Apps für einzelne Anwendungen integriert. Auch in diesem Beispiel gibt es ein Rahmenwerk (hier Toolkit genannt) für eine homogene Benutzeroberfläche in verschiedenen Teilsystemen und eine technische Schicht (Okapi Gateway), welches die Erweiterungen (Apps) integriert. In diesem Modell übernimmt VuFind die Rolle einer App und reduziert seine Aufgabe auf die Darstellung einer Katalogoberfläche.

Fazit

Die Entwicklung zu infrastrukturorientierten Open Source Lösungen weist viele Vorteile auf. Die Systemlandschaft wird konsolidiert, die Anzahl technischer Spezialexpertisen reduziert, die Serverlandschaft vereinfacht und – nicht zuletzt – den Nutzenden schlüssige Mehrwerte geboten. Gleichzeitig wird weiterhin der Fall des Vendor-Lock-ins vermieden, da die Gesamtlandschaft weiterhin aus verschiedenen Systemen besteht (nur horizontal statt vertikal getrennt) und sich die einzelne Ebene der Infrastruktur immer noch durch Apps oder Extensions ausdifferenzieren lässt.

Relativierend muss dabei gesagt sein, dass sich die Vorteile einer solchen infrastrukturorientierten Open Source Landschaft nur dann realisieren lassen, wenn die Bestandteile so lose verbunden sind, dass sie auch wirklich ausgetauscht werden können. Ansonsten werden zu stark integrierte, monolithische Systeme erzeugt, auf die gerade die individuellen Lösungen eine Antwort waren. Außerdem wird zwar z.B. die Komplexität der Serverlandschaften und zu wartenden Basistechnologien reduziert (PHP, Webserver, etc.) auf der anderen Seite gewinnt man Abhängigkeiten z.B. hinsichtlich der TYPO3-Version hinzu, wenn



die Frontends alle in demselben CMS integriert sind. Trotz dieser Einschränkungen lassen die positiven Argumente aus meiner Sicht den Schluss zu, dass sich der Trend zu den flexiblen Infrastrukturen verstärken wird. Das bedeutet gleichzeitig, dass relativ stark gekapselte Produktlösungen wie Opus als vollständiger Dokumentenserver mit einem Rückgang der Nutzung rechnen oder sich stärker auf ihre Spezialexpertise (in diesem Fall das Publikationsbackend) konzentrieren müssen, um als Baustein in der Infrastruktur einen Platz zu finden. Gleichzeitig wird sich in diesem Konsolidierungsprozess früher oder später die Frage stellen, wozu es VuFind noch braucht, wenn es TYPO3Find gibt? Oder warum es mit TYPO3Find und Kitodo.Presentation zwei Erweiterungen für die Darstellung von Objekten gibt? Der Prozess der Konsolidierung nimmt gerade erst Fahrt auf und wird noch einiges an Veränderungen bereithalten. ■

Abbildung 3:
Folio-Infrastruktur³



Prof. Dr. Matthias Finck
lehrt an der NORDAKADEMIE Usability Engineering und ist Inhaber der effective WEBWORK GmbH.

NORDAKADEMIE Elmshorn
Köllner Chaussee 11
25337 Elmshorn
matthias.finck@nordakademie.de

effective WEBWORK GmbH
Neuer Wall 18
20354 Hamburg
finck@effective-webwork.de

2 vgl. Clausnitzer, Ralf (2014): Medienübergreifende Repositorien: Mehr als „nur“ Dokumentenserver. Konferenzveröffentlichung (Vortragsfolien). 103. Deutscher Bibliothekartag, Bremen. URN: urn:nbn:de:0290-opus-15544.
3 vgl. Redewiek, Richard (2017): Folio – Einführung. VuFind Anwender-treffen 2017, Vortragsfolien, Hamburg. www.vufind.de/wp-content/uploads/2016/11/FOLIO_redweik.pdf (zuletzt besucht am 12.03.2018)