

# Ein Bibliothekssystem aus der Cloud

Das Ende der individuellen Arbeitsprozesse oder „Wie bekomme ich ein Cloudsystem in meine Infrastruktur“? – Die Eigenständigkeit in den Arbeitsprozessen bewahren und Synergien einer gemeinschaftlichen Lösung nutzen

Christian Holtkamp, Mathias Stocker, Stefan Telli

Der Umstieg vom Bibliothekssystem Aleph auf Alma in Kooperation mit SLSP<sup>1</sup> ist auch aus Sicht der Prozesssteuerung eine Herausforderung. Organisatorisch und technisch ist dabei insbesondere der Wechsel von einer On-Premises-Software (Aleph) auf eine Software aus der Cloud (Alma) besonders interessant. Im Fokus der Ausführungen stehen die Umsetzungen und Lösungen rund um die Prozesse und Workflows aus Sicht einer an SLSP teilnehmenden Bibliothek, hier der Universitätsbibliothek Bern (UB Bern)<sup>2</sup>.

Jede Bibliothek hat aufgrund eigener Strukturen individuelle Anforderungen an ein Bibliothekssystem, nicht zuletzt im Prozessmanagement. Das Bibliothekssystem soll die Prozessgestaltung unterstützen und den Rahmen zur Gestaltung der fachlichen Arbeiten und Services geben. Aber wie können diese bibliotheksspezifischen Prozesse mit einem System aus der Cloud gestaltet werden, auf das 475 an SLSP teilnehmende Bibliotheken zeitgleich zugreifen? Der Techniker würde antworten: Skalierbarkeit und individuelle Konfigurationen sind hier die Lösung. Leider funktioniert das in der Prozesssteuerung nur leidlich. Es ist verständlich, dass eine Cloudlösung mit vielen teilnehmenden Bibliotheken nie alle Besonderheiten und Charakteristika der Prozesse innerhalb jeder einzelnen Bibliothek abdecken kann. Und gerade in der Prozesssteuerung und Prozessbearbeitung zeigt sich die eigentliche Stärke dieser Cloudsysteme, nämlich die Vereinheitlichung und Standardisierung, als Schwachpunkt. Auf der einen Seite werden für viele Vorgänge und Services einheitliche Lösungen angeboten. Auf der anderen Seite können diese Lösungen leider nicht maßgeschneidert sein. Individualität und Diversität gehen verloren. Lösungen, die das Cloudsystem nicht mehr anbietet, müssen ggf. aufgegeben werden. Diese Problematik zeigt sich auch bei der IT-Infrastruktur. Wird ein cloudbasiertes Bibliothekssystem

verwendet, hat das aus Sicht der IT auf den ersten Blick große Vorteile: Es müssen keine eigenen Serversysteme (Wartung, Verfügbarkeit, Sicherheit) betreut werden. Der Zugriff erfolgt webbasiert, und i.d.R. stehen einheitliche Systeme zur Authentifizierung und Autorisierung zur Verfügung. Lösungen aus der Cloud bieten eine bessere Skalierbarkeit und versprechen eine bessere Performance. Aber auch hier gibt es gewichtige Nachteile: Man gibt bei der Verwendung eines Cloudprodukts beispielsweise die Datenhoheit (Nutzerdaten, Bibliotheksdaten) ab. Auch erscheint die Integration eines Cloudsystems in die vorhandene IT-Infrastruktur – etwa in die IT-Landschaft einer Universität – oft schwierig. Viele Prozesse werden über bibliotheksübergreifende Systeme abgewickelt. Und viele Abläufe sind von Bibliothek zu Bibliothek verschieden; etwa die Prozesse rund um den Bereich Document Delivery und Fernleihe, die Integration des eigenen Drucksystems, das Handling der Nutzerverwaltung, die Kataloganreicherung, die Integration der Lizenzverwaltung, 3rd Party-Systeme und deren Integration sowie die automatischen Prozesse zum Einspielen von Daten.

Will man die Eigenständigkeit in der Prozessgestaltung bewahren, besteht eine Lösung darin, die Abläufe an den vorhandenen Kontext anzupassen und das Bibliothekssystem aus der Cloud zu integrieren. Dies ist technisch eine große Herausforderung, da das cloudbasierte Bibliothekssystem, ebenso wie die integrierenden Systeme, i.d.R. nicht angepasst werden können. Wie könnte das Lösungsszenario also aussehen? Ein Ansatz ist die Entwicklung einer Middleware<sup>3</sup>, die mit Hilfe vorhandener Schnittstellen den ‚Glue‘ zwischen den Systemen implementiert und damit die entsprechenden Prozesse und Services ermöglicht und bereitstellt. Dabei ist es nicht das Ziel, durch diese Lösungen allgemein sinnvolle Prozesse und Normen des Netzwerkes SLSP aufzubrechen oder

1 <https://slsp.ch/de>

2 [https://www.ub.unibe.ch/ub/index\\_ger.html](https://www.ub.unibe.ch/ub/index_ger.html)

3 Middleware ist eine Infrastruktursoftware in einem verteilten System. Sie unterstützt die Verwaltung der Kommunikation der verteilten Entitäten. (Sommerville, Ian, „Software Engineering“, 10. aktual Aufl. 2018, S. 854)

zu konterkarieren, vielmehr sollen an die Bedürfnisse der einzelnen Bibliothek angepasste bzw. anpassbare Prozesse und Services zur Verfügung gestellt werden. Ziel ist es, Arbeitsabläufe zu erleichtern und gleichzeitig Vorteile im Qualitätsmanagement zu generieren. So kann die Akzeptanz der Lösung im Ganzen erhöht werden, und es werden Synergien geschaffen. Solche Lösungen können durch Adaption nicht nur partiell innerhalb einer Bibliothek, sondern auch in weiteren Institutionen genutzt werden. Denkbar wäre auch die institutionsübergreifende Bereitstellung der Lösungen als Service.

Zur Umsetzung dieser Middleware gibt es einige Ansätze. In der Praxis hat sich die Nutzung der von ExLibris in Alma angebotenen Schnittstellen bewährt. Ganz konkret ist hier die Verwendung der Alma Web API's<sup>4</sup> zu erwähnen. Im Folgenden wird an konkreten Lösungen die Umsetzung der Integration individueller Prozesse in den Alma-Kontext an der UB Bern dargestellt.

### ubBroker – eine Datendrehscheibe, viele Microservices

Während der Umstellungsphase von Aleph zu Alma wurden aus den Fachabteilungen immer wieder Wünsche an die IT der UB Bern gerichtet, verschiedenste, meistens schon vorhandene Prozesse in den Alma-Kontext zu integrieren. Dabei ging es um zentrale Services wie die Integration eines eigenen Druckdienstes, die Prozesssteuerung im Bereich Document Delivery, die automatische Verarbeitung von Katalogdaten oder die Bearbeitung von Nutzerdaten. Diese Anfragen führten zur Überlegung der Entwicklung eines Backends für die flexible und performante Bearbeitung und Weiterleitung von eingehenden Nachrichten bzw. Events an weitergehende Services wie z.B. Druck, E-Mail oder Webservices. Dafür soll eine zentrale Lösung inklusive Infrastruktur zur automatischen Verarbeitung, Routing und Integration hybrider Daten aus unterschiedlichen Systemen in unterschiedliche Systeme bereitgestellt werden. Eine zentrale Instanz übernimmt die Filterung, die Aufbereitung, das Routing und die Integration der Daten. Diese 'Datendrehscheibe' soll möglichst universal, beziehungsweise generisch gestaltet werden, damit mit möglichst geringem Aufwand verschiedenste Anwendungsszenarien über entsprechende Microservices abgebildet werden können. Daten werden also von einem Producer bereitgestellt und laufen über einen Vermittlungsdienst, den Broker. Hier werden die Daten gefiltert, verarbeitet und geroutet, um anschließend

über Microservices im Bereich des Consumers an die Zielinstanz übergeben zu werden. Folgende Anforderungen erscheinen dabei wichtig:

- Universaler Datenimport
- Generische Verarbeitung der Daten
- Transaktionssichere Verarbeitung der Daten
- Konfigurierbares Routing der Daten
- Generische Exportfunktionen für die Daten
- Prozesse müssen automatisiert ablaufen

#### Architektur

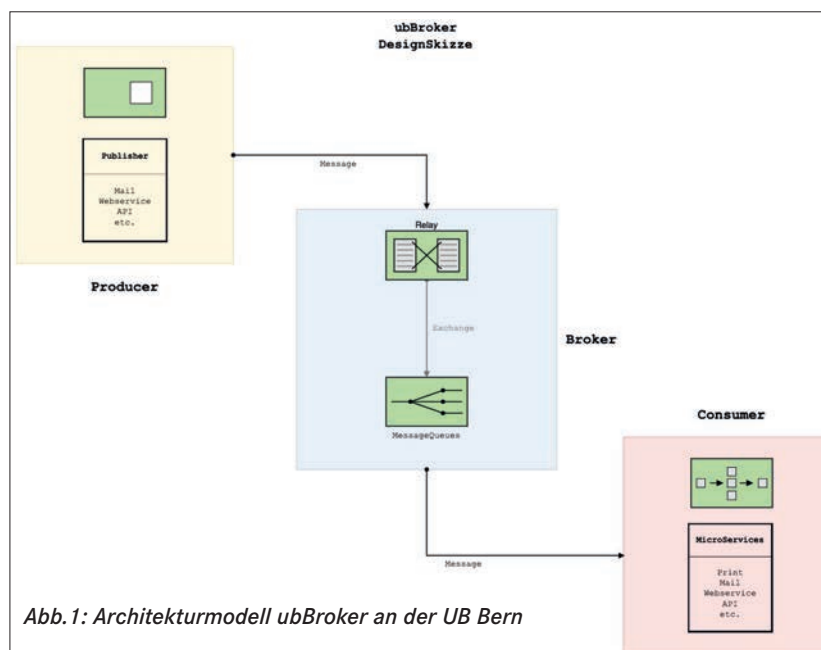


Abb. 1: Architekturmodell ubBroker an der UB Bern

#### Producer

Der Producer ist für die Bereitstellung der Daten zuständig. Entweder übergibt er die Daten aktiv an die Vermittlungsservices, oder die Vermittlungsservices holen sich die Daten ab.

#### Exchange

Der Vermittlungsdienst, in dem eine Message erzeugt und an die passende MessageQueue<sup>5</sup> übergeben wird. Hier kann z.B. die Verteilungslogik implementiert sein.

#### Broker

Der Broker übernimmt die transaktionssichere Übertragung und Pufferung der einzelnen auszutauschenden Daten. Hier werden die MessageQueues bereitgestellt. Aus diesen MessageQueues bedienen sich die unter Consumer beschriebenen Microservices, die dann zum Beispiel das eigentliche Einspielen der Daten in Alma übernehmen oder einen Druck

<sup>4</sup> <https://developers.exlibrisgroup.com/alma/apis/>

<sup>5</sup> In der Informatik sind Message Queues Kommunikationsprotokolle, die für die Interprozesskommunikation oder für die Inter-Thread-Kommunikation innerhalb desselben Prozesses verwendet werden. siehe: [https://de.wikipedia.org/wiki/Message\\_Queueing](https://de.wikipedia.org/wiki/Message_Queueing)

anstoßen. Die MessageQueues werden in die Infrastruktur des ubBrokers bei der UB Bern integriert. Hier steht mit RabbitMQ<sup>6</sup> als Message Broker ein System zur Verfügung, welches das AdvancedMessageQueueingProtocol (AMQP) implementiert und damit die Voraussetzungen erfüllt. Hier werden jeweils pro Microservice MessageQueues konfiguriert, die dann die entsprechenden Consumer bedienen.

#### Consumer

Im Bereich des Consumers werden die einzelnen Microservices implementiert, die für die Verarbeitung der Daten zuständig sind. Gesteuert werden diese über die MessageQueues im Broker. Jede MessageQueue im Broker ist mit einem Microservice im Bereich des Consumers verbunden.

### Anwendungsfall: Kataloganreicherung (CE) mit Alma

Wie kann man sich nun eine konkrete Umsetzung eines Microservices im ubBroker vorstellen? Als Beispiel wird im Folgenden eine zentrale, kooperative Lösung zur Integration der Kataloganreicherung in den SLSP Katalog dargestellt.

Der Workflow entspricht einem klassischen Eingabe-Verarbeitung-Ausgabe-Prozess. Es werden Daten zur Kataloganreicherung bereitgestellt. Diese werden gefiltert, verarbeitet und geroutet. Und schließlich werden die Daten in den SLSP-Alma Katalog eingespielt. Im konkreten Fall stellt der Producer die Daten (digitale Titelseiten, Inhaltsverzeichnisse, Cover und/oder Klappentexte inklusive OCR-Daten) für jeden Titel bereit. Zusätzlich wird zur Übergabe der Daten an den Vermittlungsdienst noch eine Interface-Datei pro Titelaufnahme benötigt (CE-DataPackage). Im CE-DataPackage stehen Metadaten des Titels zur Zuordnung sowie Zugriffsdaten für die bereitgestellten Daten zur Kataloganreicherung eines Titels. Der Vermittlungsdienst wertet die CE-DataPackages aus. Hier wird dann die Filterung, gegebenenfalls Bearbeitung sowie das Routing der

Daten in den jeweiligen Bibliotheken zugeordneten MessageQueues durchgeführt. Über die jeweiligen MessageQueues werden Microservices auf Seiten des Consumers aufgerufen, die dann die Daten, idealerweise über die Alma-Webservices, in Alma einspielen.

#### Architektur

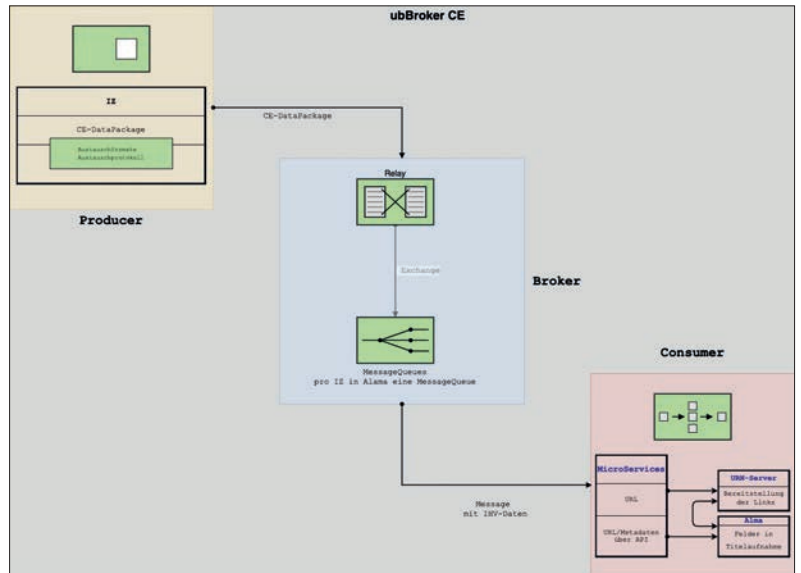
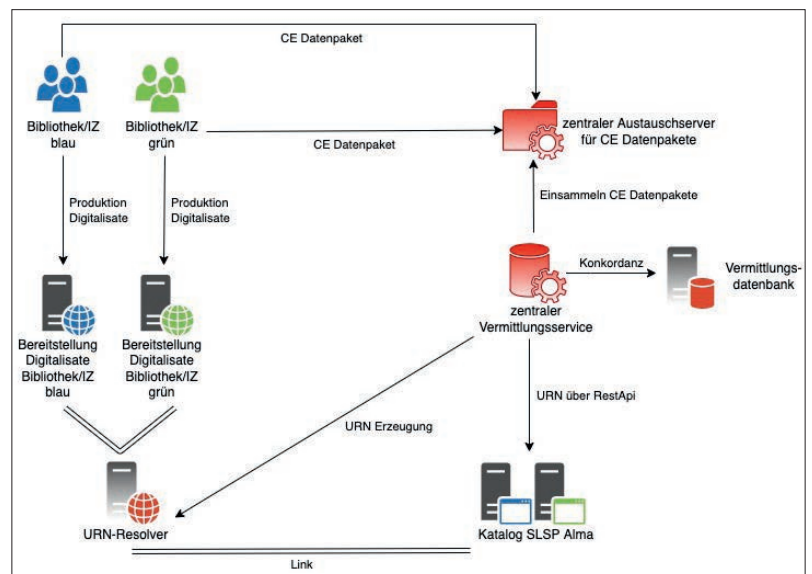


Abb. 2 und 3: Infrastruktur und Architekturmodell ubBroker Kataloganreicherung



#### Producer

Im Kontext der Kataloganreicherung ist der Producer für die Bereitstellung der Daten zur Kataloganreicherung zuständig. Konkret werden hier von den teilnehmenden Bibliotheken für die einzuspielenden Titelaufnahmen die CE-DataPackages erstellt und zur weiteren Verarbeitung freigegeben. I.d.R. sind die

<sup>6</sup> RabbitMQ ist eine Open Source Message Broker Software, die das Advanced Message Queuing Protocol implementiert. Siehe <https://de.wikipedia.org/wiki/RabbitMQ>



Massmann –  
seit mehr als 30 Jahren  
Ihr zuverlässiger Partner  
für Bücher und  
eBooks



Massmann Internationale Buchhandlung  
Luruper Chaussee 125  
22761 Hamburg  
Telefon 040/7670040  
Telefax 040/76700410  
E-Mail [info@massmann.de](mailto:info@massmann.de)  
Internet [www.massmann.de](http://www.massmann.de)

Bibliotheken für die Bereitstellung selbst verantwortlich. Im Rahmen dieser Lösung wird von der UB Bern das CE-DataPackage als Austauschformat und ein Austauschservice vorgeschlagen.

Format CE-DataPackage:

- Einfaches XML-Format

Inhalt CE-DataPackage:

- Grunddatensatz aus der Titelaufnahme zur Identifizierung des Titels
- Eindeutiges Kennzeichen, Bezeichner und Stammdaten (inklusive Benachrichtigungsoptionen) der liefernden Bibliothek
- Zu importierende Dokumententypen
- Links auf die digitalen Daten für die Kataloganreicherung, also z.B. auf das durchsuchbare PDF des Inhaltsverzeichnisses eines Titels

Austauschservice:

Für den ersten Schritt sollte ein Austausch der CE-DataPackages über das Filesystem ausreichen. Entweder erfolgt ein Upload der Packages der Bibliotheken auf ein zentrales Verzeichnis bei der UB Bern oder die teilnehmende Bibliothek stellt für den ubBroker ein entsprechendes Download-Verzeichnis zur Verfügung.

*Exchange*

Der Vermittlungsdienst im Kontext der Kataloganreicherung sammelt die CE-DataPackages ein. Der Dienst prüft und wertet die Pakete pro Titelaufnahme aus, ggf. muss hier noch eine Datenanreicherung durchgeführt werden. Über die Alma RestfulAPI sollte eine direkte Dublettenprüfung in Abhängigkeit des Dokumententyps möglich sein. Wird eine Dublette erkannt, wird eine entsprechende Benachrichtigung an die liefernde Bibliothek erstellt. Sind alle Daten eines CE-DataPackages ausgewertet und bearbeitet, wird pro zu importierendem Dokumententyp eine Message erzeugt und an die passende MessageQueue übergeben. Der Vermittlungsdienst zur Kataloganreicherung wird in die vorhandene ubBroker-Infrastruktur der UB Bern integriert. Hierzu werden die Funktionen zur Kataloganreicherung (Einsammeln, Kontrolle, Datenanreicherung und Dublettenprüfung des CE-DataPackages und Routing in entsprechende MessageQueues) implementiert. Für jede teilnehmende Bibliothek wird ein Service konfiguriert, der für die jeweilige Datenverarbeitung, die Erzeugung der Message und das Routing in die entsprechende MessageQueue zuständig ist.

*Broker*

Der Broker übernimmt die transaktionssichere Übertragung und Pufferung der einzelnen in Alma zu importierenden Dokumententypen (Titelseiten, Inhaltsverzeichnisse, Cover und/oder Klappentexte) im Kontext einer Titelaufnahme. Hier werden pro Bibliothek und pro Dokumententyp MessageQueues bereitgestellt. Aus diesen MessageQueues bedienen sich die unter Consumer beschriebenen Microservices, die dann das eigentliche Einspielen der Daten in Alma übernehmen. Hier werden jeweils pro Bibliothek und pro Dokumententyp MessageQueues konfiguriert, die dann die entsprechenden Consumer bedienen.

*Consumer*

Im Kontext der Kataloganreicherung und Einspielung der Daten in den SLSP Katalog werden folgende Funktionen in den Microservices benötigt:

- Pro Dokumententyp und Titelaufnahme muss eine URN für die jeweiligen Digitalisate erzeugt und in einen URN-Resolver eingetragen werden
- Speicherung der Titeldaten/URLs/URN in eine Datenbank
- Pro Dokumententyp und Titelaufnahme Import der URN in den SLSP Katalog über die Rest API von Alma

Die MessageQueues für die Kataloganreicherung im RabbitMQ übergeben die Messages an die jeweils zuständigen Microservices. Pro Bibliothek und je Dokumententyp gibt es einen Microservice. Dieser erzeugt eine URN, übergibt sie an den URN-Resolver, speichert die Titeldaten in einer Datenbank und spielt die URN über die RestAPI von Alma in das passende Feld im SLSP Katalog. Eine Rückmeldung über die Einspielung kann via Journal oder E-Mail erfolgen.

Die Anbindung der Kataloganreicherung an Alma ist nur ein Beispiel für die Verwendung des ubBrokers als Datendrehscheibe. Inzwischen wurden diverse weitere Microservices implementiert, wie z.B. für die Integration des Drucksystems oder die Anbindung der 3rd Party-Applikation zur Bearbeitung von Document Delivery Bestellungen an Alma. Weitere Integrationen und Services durch die IT der UB Bern sind bereits in Planung.

### **ubAuth – Authentication Server der UB Bern**

Ein weiteres Beispiel für die Integration eigener Prozesse ist die Implementierung eines eigenen Authentifizierungs-Servers bei der UB Bern. Die IT der UB Bern betreibt diverse Anwendungen, welche eine Authentifizierung der Nutzenden benötigen, wie z.B.:

- In die Universitäts-Website eingebettete Formulare (beispielweise Anschaffungsvorschläge, Kopienbestellung)
- Reservations-Anwendungen (beispielweise Gruppenräume, Semesterarbeitsplätze)
- OCLC EzProxy (Zugriff auf lizenzierte Verlagsinhalte aus dem Uni Netzwerk und für Einwohner des Kantons Bern)
- Unpersönliche Recherche-Stationen (Anmeldung am Proxy-Server vor Nutzung Internet; Vorgabe der Universität „Keine unpersönliche Nutzung des Uni-Netzwerks“)
- IIF Server
- Diverse Anwendungen für die Nutzenden und Mitarbeitenden der UB Bern

Vor der Einführung von SLSP und der SWITCH edu-ID<sup>7</sup> nutzten diese Anwendungen die unterschiedlichsten Protokolle und Formate zur Authentifizierung:

- Campus Account (LDAP oder ADFS/SAML)
- Bibliothekslögin (Aleph X)
- SWITCH aai (Shibboleth/SAML)

Das hat oft zu Verwirrung bei den Nutzenden geführt, da nicht immer klar war, welches Login (Benutzername und Passwort) nun zu verwenden ist. Die Einführung der SWITCH edu-ID nur für das Discovery hätte diese Verwirrung wohl noch vergrößert. Zudem musste sichergestellt werden, dass Anwendungen, die benutzerbezogene Daten aus dem Bibliothekssystem abfragen (Bibliothekslögin/Aleph X), das auch mit Alma können. Um die Nutzung und Wartung der Anwendungen sowie die Einführung von SLSP mit der edu-ID an der UB Bern zu vereinfachen, wurde entschieden, dass ein eigener Authentifizierungs-Server als Drehscheibe für alle Anwendungen

aufgesetzt wird. Der Server kommuniziert mit den Anwendungen über ein einheitliches Protokoll in einem einheitlichen Format (OIDC: OpenID Connect<sup>8</sup>). Die Anwendung meldet den Bedarf an benutzerbezogenen Daten über den erlaubten Anwendungsbereich (Scope). Die Nutzenden willigen in die Übermittlung dieser Daten ein. Je nach Scope reichert der Authentifizierungs-Server die Benutzungsdaten aus den Umssystemen (Alma, Active Directory der Uni) an. Neben der Anmeldung mit der edu-ID wurden noch weitere Szenarien umgesetzt, ein Beispiel dafür ist die Anmeldung mit einem Voucher (Code mit begrenzter Gültigkeitsdauer). In den Bibliotheken befinden sich unpersönliche Recherchestationen. Wer noch keinen edu-ID Account hat, erhält gegen Vorlage eines amtlichen Ausweises an der Theke einen solchen Voucher und kann sich dann an einer Recherche-Station einen edu-ID Account anlegen und sich für SLSP registrieren. Somit ist auch die Vorgabe der Universität „Kein unpersönlicher Zugang zum Uni Netzwerk“ gewährleistet. Für die Anwendung selbst spielt es keine Rolle, welches Szenario verwendet wird, da sie die Benutzungsdaten vom Authentifizierungs-Server über OpenID Connect in einem einheitlichen und vordefinierten Format erhält.

Systemübersicht

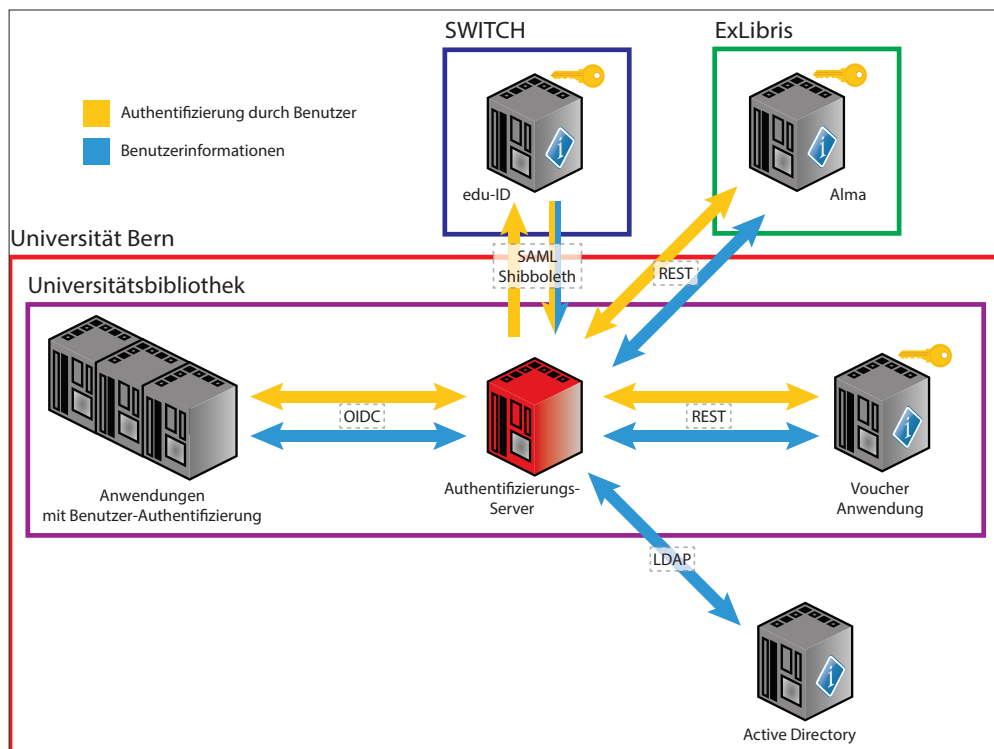


Abb.4: Systemübersicht: Authentifizierungsserver UB Bern

7 SWITCH edu-ID ist eine persönliche elektronische Identität und erlaubt den Zugriff auf alle föderierten Dienste in der Schweiz. Siehe <https://help.switch.ch/eduid/faqs/?lang=de#eduid-about>

8 OpenID Connect (OIDC) ist eine Authentifizierungsschicht, die auf dem Autorisierungsframework OAuth 2.0 basiert. Siehe [https://de.wikipedia.org/wiki/OpenID\\_Connect](https://de.wikipedia.org/wiki/OpenID_Connect)

Die Einführung einer einheitlichen Authentifizierungsmethode für die Anwendungen der Universitätsbibliothek hat sich bewährt. Wo vorher bei neuen Anwendungen relativ viel Zeit in die Anbindung an entsprechende Authentifizierungssysteme aufgewendet wurde – insbesondere auch Wartezeit bezüglich Einrichtung bei Dritten (Informatikdienste der Uni, SWITCH) –, kann nun die Authentifizierung meist in kurzer Zeit eingerichtet und implementiert werden. Weitere Anwendungen sind bereits in Planung, so z.B. die Anzeige der eigenen offenen Ausleihen in der CampusApp der Universität Bern.

### Fazit

Die beschriebenen Lösungen zeigen, dass der Einsatz eines Bibliothekssystems in der Cloud nicht zwangsläufig den Verlust der individuellen Arbeitsprozesse und Lösungen mit sich bringt. Es ist jedoch immer eine Herausforderung, das System in die eigene Infrastruktur zu integrieren und individuelle Workflows zu implementieren. Dabei ist darauf zu achten, dass Nutzen und Aufwand ausgewogen sind. Der Aufwand kann durch die synergetische Nutzung positiv beeinflusst werden. Wichtig erscheint in diesen Szenarien die gezielte Zusammenarbeit aller an solchen Lösungen interessierten Stakeholder. ■



b  
UNIVERSITÄT  
BERN

### Über die UB IT

Die Informatikabteilung der UB Bern betreibt und entwickelt eigene Anwendungen im Bereich Bibliotheks-informatik. Daneben werden auch Lösungen von Drittanbietern eingesetzt und betreut. In Bern ist die Abteilung zudem Ansprechpartnerin bei technischen Fragen rund um Alma und Primo VE.

Die Informatikabteilung der UB Bern ist organisatorisch den zentralen Diensten der UB zugeordnet. Gleichzeitig bildet sie die Schnittstelle zu den zentralen Informatikdiensten der Universität.



#### Christian Holtkamp

Applikationsentwickler  
Universität Bern,  
Universitätsbibliothek  
christian.holtkamp@unibe.ch



#### Mathias Stocker

Applikationsentwickler  
Universität Bern,  
Universitätsbibliothek  
mathias.stocker@unibe.ch



#### Stefan Telli

Leiter Informatik  
Universität Bern,  
Universitätsbibliothek  
stefan.telli@unibe.ch