Projektskizze Semesterprojekt CAS Big Data

Weiterentwicklung ausgewählter Komponenten der Informationsplattform swissbib als Grundlage für Dienstleistungen wissenschaftlicher Bibliotheken im Jahre 2020+

Günter Hipler

1. Umfeld / Ausgangslage

Das Projekt swissbib an der Universitätsbibliothek Basel ist eine im Rahmen des Programms SUK-P2[1] durch swissuniversities geförderte öffentliche Serviceplattform für Daten- und Suchdienste. Die bibliographischen Metadaten aller schweizerischen Universitätsbibliotheken, der Nationalbibliothek, einzelner Kantonalbibliotheken, diverser Dokumentenrepositories sowie zusätzlichen Spezialsammlungen z.B. aus Archiven werden in einem Datenhub aufbereitet [2]. Dieser Datenhub ist Grundlage für interaktive Services (Information Retrieval durch Forschende, Studenten und die interessierte Öffentlichkeit)[3] oder APIs, die wiederum von anderen services für ihre Zwecke genutzt werden können. Durch die Aufbereitung im Datenhub werden ca. 35 Millionen Datensätze der einzelnen Institutionen mittels Deduplizierungsmechanismen auf ca. 21 Millionen Aufnahmen zusammengefasst. Diese 21 Millionen Entities werden über Clusteringverfahren (Zusammenfassung von ähnlichen Aufnahmen wie zum Beispiel mehrere Auflagen eines Werkes) sowie Verlinkung und Anreicherung mit externen Objekten nochmals veredelt.

Diese bibliographischen Beschreibungen, aktuell mehrheitlich physikalische Medien, werden in Zukunft verstärkt durch Beschreibungen digital zugänglicher Objekte ergänzt. Dies werden zu Beginn vor allem strukturierte Metadaten von Artikeln sein [4], weniger strukturierte Beschreibungen wie beispielsweise Forschungsdaten sind jedoch nicht ausgeschlossen.

Die swissbib Plattform ist damit für den Einsatz von Methoden und Verfahren aus dem Forschungsbereich “Big Data” sehr geeignet. Die häufig genannten Kriterien für eine Definition von Big Data sind anwendbar:

1) Volume: 35 Millionen initiale Objekte sind sicherlich eine Einstiegsgrösse für “BigData”. Mit zusätzlichen Artikeldaten kann die Grenze von 100 Millionen schnell überschritten werden. Werden die bisherigen klassischen dokumentenorientierten Beschreibungen nach den Methoden des semantischen Webs zur besseren Verlinkung in einzelne Konzepte aufgebrochen[5], ist bereit jetzt die 100 Millionen Grenze erreicht [6]

1) Für weitere Informationen vgl. https://www.swissuniversities.ch/de/organisation/projekte-und-programme/suk-p-2-wissensch-information-zugang-verarbeitung-speicherung/ https://www.swissuniversities.ch/fileadmin/swissuniversities/Dokumente/DE/UH/SUK_P-2/Abstract_swissbib.pdf
2) Für eine Zusammenstellung der beteiligten Institutionen: https://www.swissbib.ch/Libraries
3) https://www.swissbib.ch
4) vgl. die aktuelle Zusammenarbeit mit dem Projekt Nationallizenzen http://swissbib.blogspot.ch/2016/03/national-licences-and-article-metadata.html
5) Dies ist Aufgabe des Partnerprojekts http://linked.swissbib.ch Weitere Informationen dazu finden sich u.a. in verschiedenen Blogbeiträgen: http://swissbib.blogspot.ch/2014/06/considerations-for-development-of.html http://swissbib.blogspot.ch/2016/04/swissbib-data-goes-linked-teil-1.html http://swissbib.blogspot.ch/2016/04/swissbib-data-goes-linked-teil-2.html
6) vgl. http://lodsearch.swissbib.ch/testsb/_search

2) Velocity: Die Daten der swissbib Plattform sind in ständiger Veränderung. Eine Initialisierungsphase, bei der Entities aus den aktuell rund 20 verschiedenen Datenquellen geladen und aufbereitet werden (Deduplizierung, Clustering, Anreicherung) benötigt mit der aktuellen Technologie nicht unter 14 Tage. Nach der Initialisierung wird der Datenhub mit den Ergebnissen der täglichen Arbeit der Bibliothekar-Innen aus den verschiedenen Quellen jede Nacht aktualisiert. Im Tagesdurchschnitt sind dies zwischen 100.000 und 300.00 Messages. Der aktuelle Datenhub, dessen Storagesystem auf relationaler Datenbanktechnik beruht, benötigt hierfür in der Regel 2 bis 6 Stunden.

3) Variety: Metadaten sind per Definition relativ stark strukturierte Daten. Allerdings gibt es alleine im Bibliotheksbereich eine Vielzahl unterschiedlicher Formate, auch wenn aktuell der noch traditionelle MARC Standard[7] vorherrschend ist. Andere Formate, die sich auch in der Struktur unterscheiden, müssen jedoch berücksichtigt werden. So sind bspw. Archivdaten stark hierarchisiert während im Bibliotheksbereich eher flache Strukturen vorherrschend sind. Durch die zunehmende Verlinkung sind weitere Formate von hoher Relevanz (bspw. RDF mit seinen unterschiedlichen Serialisierungen[8] oder Beacon[9]). Potentielle Forschungsdaten vergrössern die 'Variety' nochmals. Es werden also Verfahren benötigt, die zumindest semi-struktierte Daten verarbeiten können.

4) Veracity: Hierunter wird häufig die Richtigkeit und Echtheit der Daten zusammengefasst [10] Während die Echtheit der Daten aus dem institutionellen Umfeld i.d.R. wohl kaum angezweifelt werden muss, ist die Richtigkeit schon eher fraglich. Die traditionelle Erstellung von Metadaten ist “Menschenwerk” und unterliegt damit einer intellektuellen Kontrolle. Die Beschreibungen digitaler Objekte (bspw. Artikeldaten) sind jedoch immer mehr das Ergebnis maschineller Prozesse. Verfahren sollten deshalb Möglichkeiten kennen, die Richtigkeit anhand von ausgewählten Kriterien zu analysieren.
Neben der Richtigkeit von Daten ist das Thema Lizenzierung im Bereich Metadaten aktuell von grösserer Bedeutung. Kaufen Bibliothen sog. Fremddaten als Ergebnis maschineller Prozesse (bspw. von Verlagen) und vermischen diese mit Daten aus anderen Quellen, stellt sich die Frage, in welcher Form das gesamte Dataset anderen services zur weiteren Nutzung bereitgestellt werden kann[11].
BigData sollte damit Mechanismen ermöglichen, mit denen evtl. nur Teile eines gesamten datasets anderen Diensten zur Weiternutzung angeboten werden.

2. Problemstellung

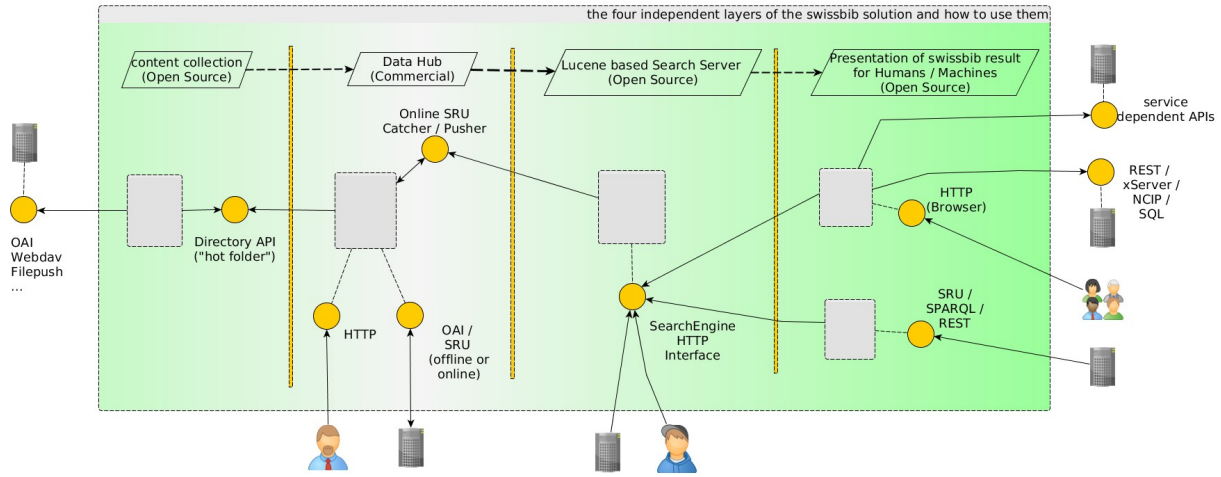
Der Aufbau der aktuellen swissbib Plattform, die sich seit Februar 2010 im produktiven Einsatz befindet, folgt einem konsequenten Architekturmuster. Die eingesetzten Komponenten
- Data ingestion und pre-processing (content collection)
- Data management und enrichement (data hub)
- Document processing und Search engine
- Komponenten für Benutzer- und Maschinenservices

sind voneinader unabhängig in einzelnen Layern angeordnet und kommunizieren ausschliesslich über Schnittstellen miteinander. Diese Schnittstellen sind nicht nur intern verfügbar sondern werden grösstenteils auch externen Services bereitgestellt. Dies ermöglichte im Verlaufe der Zeit den Austausch einzelner Komponenten, nachdem sich die äusseren Umstände verändert hatten. So wurde aus einer nahezu 100% kommerziellen Lösung eine Plattform, die zu einem grossen Teil auf Open Source Komponenten beruht[12] Der Leitgedanke von swissbib ist nicht von einem sich gegenseitig

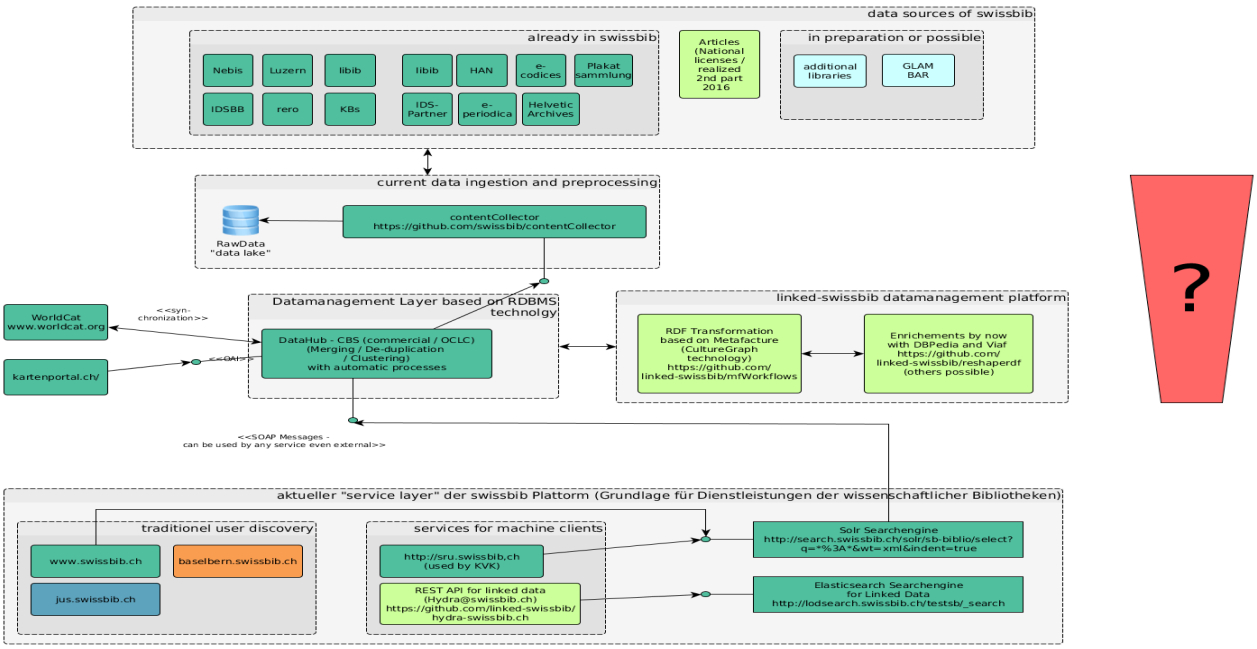
7) Vgl. https://www.loc.gov/marc/
8) https://www.w3.org/RDF/
9) https://meta.wikimedia.org/wiki/Dynamic_links_to_external_resources
10) vgl. https://datafloq.com/read/3vs-sufficient-describe-big-data/166
11) Daten sollten wenn immer möglich unter einer CC0 Lizenz bereitgestellt werden. https://creativecommons.org/publicdomain/zero/1.0/deed.de
12) s. auch https://github.com/swissbib/ und https://github.com/linked-swissbib Ausgetauscht wurden vor allem die layer der Suchmaschine (von Fast zu SOLR), der Präsentationskomponente (kommerzielles

ausschliessenden “make or buy” sondern von einem sich je nach den Umständen ergebenden “make and buy” geprägt. Dies machte den service in einer sich permanent verändernden Umwelt sehr flexibel und agil.

Dieses ursprüngliche Design ist in dem nachfolgenden Schaubild abgebildet. Es wurde bereits im Jahre 2013 im Rahmen einer Präsentation vorgestellt[13]



Diese Anpassbarkeit bewährte sich auch in der letzten grösseren Erweiterung seit Beginn des Jahres 2015 im Rahmen des SUK-P2 Projekts linked-swissbib[14] Es mussten Komponenten gefunden werden, die den bestehenden Datenhub um die Möglichkeiten erweiterten, bibliographischen Metadaten in das RDF Format zu transformieren und die so entstehenden neuen Ressourcen mit externen Ressourcen zu verlinken. Hierfür verwenden wir das an der Deutschen Nationalbibliothek im Rahmen des Projekts Culturegraph entwickelte Framework Metafacture[15]. Es baut auf einer datenflussorientierten Domain Specific Language (Metamorph)[16] auf und benutzt Pattern zur Datenanalyse, die denen von Hadoop mit Map/Reduce ähnlich sind. Dadurch ist eine Transformation hin zu einem batchorientierten Datencluster möglich und wurde im Projekt Metafacture-cluster[17] auch umgesetzt. Bisher jedoch nur im Rahmen eines Prototyps. Die nachfolgende Abbildung verdeutlicht die Integration der linked-data Plattform.



In der erweiterten Infrastruktur ist der Layergedanke mit in den Layern gekapselten Komponenten noch deutlich sichtbar. Der Gedanke des Datenlayers akzentuiert sich jedoch stärker und der vorher eher eigenständige Suchlayer wird allmählich zum Teil eines 'Service Layers'.

Welche Probleme lassen sich in der aktuellen Architektur finden?

1) Von Beginn an setzen wir beim Datenhub auf die CBS genannte Komponente der Fa. OCLC. Diese wird vor allem bei grösseren Verbünden als zentraler Katalog (interaktiver Nutzerbetrieb) sowie Managementlösung für Metadaten eingesetzt[18]. In swissbib kennen wir keinen Nutzerdialog, die von uns genutzen Prozesse zur Deduplizierung, Anreicherung und Clustering sind automatisiert und batchbasiert. Mit der Komponente CBS steht durch den bereits langen Betrieb und die immer noch intensive Nutzung in grossen Verbünden eine Menge KnowHow zum Datenmangement bereit. Die Verkettung einzelner Komponenten auf der OS-Ebene ergibt vielfältige Möglichkeiten für Datentransformationen. Storagesystem ist eine relationale Datenbank (Sybase, von der Leistungsfähigkeit mit Oracle vergleichbar). In dieser Kombination bietet das System für das Datenmanagement leistungsfähigere und flexibere Möglichkeiten für NutzerInnen als heute verfügbare cloudbasierte Systeme von Firmen wie ExLibris[19] oder OCLC[20] Während OCLC seinen grossen Datenpool Worldcat auf Hadoop und HBase gestützte Verfahren umgestellt hat[21] sind mir bisher keine Informationen zugänglich, dass dies bei anderen Firmen in ähnlicher Weise fortgeschritten ist. Ich vermute, dass dort die relationale Technik noch vorherrschend ist.

Obwohl CBS als das aktuelle 'Arbeitspferd' im Bereich data management für swissbib als 'gesetzt gilt', ist die Perspektive eines end-of-live absehbar. Zum einen, weil der Hersteller OCLC KundInnen verstärkt auf seine cloud-basierte Lösung lenken möchte und deshalb die Maintenance ab einem jetzt nicht definierten Zeitpunkt einstellen wird, zum anderen weil traditionelle relationale Techniken im heutigen Umfeld wie beschrieben an Grenzen stossen (werden)[22] Das swissbib Team stellt sich auf eine Perspektive von 5 bis max. 10 Jahren ein, innerhalb der ein produktiv nutzbarer Ersatz für die Data-Hub Funktionalität gefunden sein sollte.

2) Die von uns contentCollector genannte Komponente für das 'data ingesting' erfüllt zwar momentan sämtliche Anforderungen und ist flexibel, weist jedoch Schwachstellen auf, die vor allem in der mittleren Zukunft Probleme bereiten können.
- Es ist eine Eigenentwicklung, die entsprechende Maintenance verlangt. Anders als bei sich häufig ändernden user services, die hohe Agilität verlangen, sollte ein Backend-Service für mich vor allem eine hohe Funktionsstabilität aufweisen. Ausserdem muss er sich gut in das weitere Umfeld integrieren können.
- Werden die Datenmengen signifikant vergrössert, wird die Komponente an ihre Leistungsgrenzen stossen oder müsste durch parallel laufende Instanzen skaliert werden. All die dafür notwendigen Massnahmen müssen jedoch eigenständig entwickelt und aufgesetzt werden.
- die Komponente benutzt eine NoSQL Datenbank (MongoDB), in der Rohdaten vor der Verarbeitung im Datenhub abgelegt werden. Dieser storage kann im weitesten Sinne als „data-lake“ bezeichnet werden und hat sich im Produktionsprozess bereits sehr oft als hilfreich erwiesen. Die Nutzung dieser Rohdaten vor allem durch die Datenhubkomponente könnte jedoch enger und häufiger sein.
- Der für mich entscheidende Punkt: es gibt heute OpenSource Komponenten, die diese Schwachstellen gut fokussieren und die zukünftigen Anforderungen umsetzen würden.

3) Aus den Abbildungen lässt sich ein wenig die Herkunft des klassischen Datenhub als universelle und generelle Serviceplattform für Mensch und Maschine erkennen. Er bietet maschinelle (OAI / SRU / SOAP) sowie interaktive Schnittstellen (HTTP) an. Ebenso ist die Synchronisation von Daten mit

18) https://www.oclc.org/de-DE/publications/newsletters/enews/2013/33/de-04.html
19) http://www.exlibrisgroup.com/category/AlmaOverview
20) https://www.oclc.org/worldshare.en.html
21) https://www.oclc.org/news/releases/2013/201329dublin.en.html
22) vgl. dazu die Anmerkungen zu Velocity im Abschnitt Umfeld / Ausgangslage

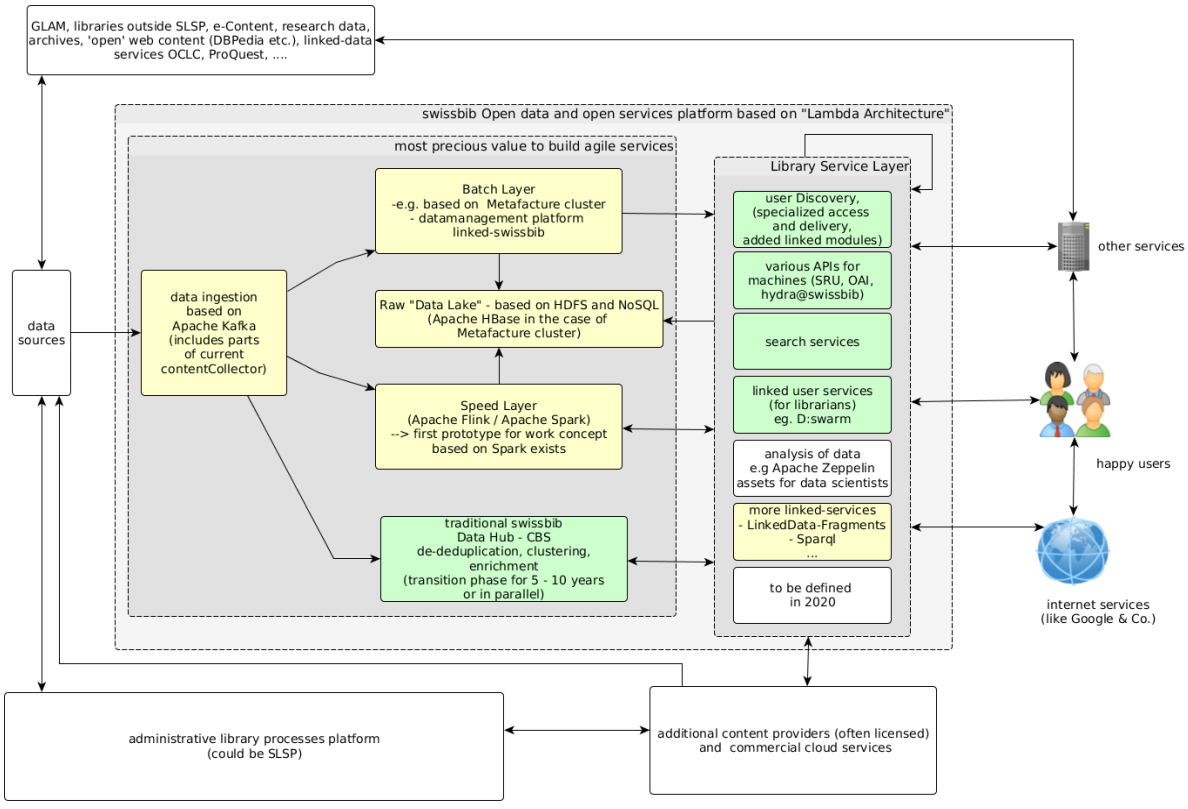
externen Plattformen[23] möglich. Heute würde man solche services eher in einen eigenen Servicelayer auslagern und die Datenkomponente nicht darauf spezialisieren.

4) In der Entwicklung der Datenmanagementplattform für verlinkte Daten haben wir momentan das Problem, dass wir den klassischen Datenhub als Basis benutzen. Die Verfahren basieren jedoch auf sehr unterschiedlichen Technologien und wir kennen die Algorithmen der De-Duplizierung sowie des Clustering nur bedingt, da es sich bei CBS um ein kommerzielles System handelt, bei dem der Zugriff auf die Methodik und den Code nur bedingt möglich ist. Es wäre für uns sehr viel einfacher und zielführender, wenn wir die Verfahren auf Basis moderner Softwaretechnolgien, wie wir sie bereits in der neuen linked-swissbib Plattform einsetzen, zusammenfassen könnten.

3. Lösungsansatz

Die in 2 dargestellten Probleme lassen sich auf Basis einer in der Literatur als „Lambda Architektur“[24] bezeichneten Modells unter Einsatz von Komponenten aus dem Big Data Umfeld lösen.

Die nachstehende Abbildung soll helfen, diesen Ansatz zu verdeutlichen.



1) Der aktuelle contentCollector kann durch Apache Kafka ersetzt werden. Die Architektur von contentCollector basiert auf sog. pipes für die verschiedenen Datenquellen. Implementiert sind diese in Python. Ich gehe davon aus, dass sich der Kern dieser pipes als Kafka-clients (sog. Producer)

23) www.worldcat.org
24) vgl. http://lambda-architecture.net/

wiederverwenden lassen und den broker-Kern mit messages feeden können.

2) Als zentrale Komponente des 'Batch Layer' sehe ich im Moment vor allem das bisher als Prototyp vorliegende Repository Metafacture-cluster. Es basiert auf klassischen Map/Reduce Jobs und verwendet als storage die Apache NoSQL Datenbank Hbase.

Im Rahmend des BatchLayers ist auch der Einsatz des Backends der Datenmanagementplattform D:Swarm denkbar[25]. Diese Plattform benutzt für den storage die Property Graphendatenbank Neo4J.

3) Für den sog. 'Speed Layer' sehe ich die beiden zur Zeit bekanntesten Vertreter des Online Stream Processing Apache Flink / Apache Spark

4) An die neue data ingest Komponente liesse sich ohne grösseren Aufwand durch Entwickllung eines zwischenzuschaltenden Consumers die bisherige Datenhubkomponente CBS für eine Übergangsphase integrieren. Damit könnte der bisherige produktive Betrieb bis zur Fertigstellug und Reife der neuen Komponenten weiterbetrieben werden.

5) Der neue 'Library Service Layer' kann mit vielen bereits jetzt bestehenden services gefüllt werden. (Search Services, Discovery, diverse APIs, …). Vor allem bereits jetzt bestehende Suchservices (Solr, Elasticsearch) lassen sich durch die gute Kombinierbarkeit mit Storagesystemen (z.B. HDFS) oder Streaming Komponenten (Flink / Spark) in der Funktionalität gut erweitern aber auch optimieren.

Da dieser Layer die Angebote wissenschaftlicher Bibliotheken beinhaltet, werden sich diese services am schnellsten im Verlaufe der Zeit ändern (müssen). Dies bedingt die Einbettung in das sich permanent wandelnde Umfeld globaler Internetplayer, kommerzieller (häufig konkurrierender) Dienste sowie die Erwartungen der stakeholder wissenschaftlicher Bibliotheken. Die auf Basis der Lambda Architektur neu gestaltete Plattform zur Analyse und Verarbeitung von Informationen bietet für die Gestaltung solcher Dienste sehr gute Voraussetzungen.

4. Arbeitspakete der Projektarbeit sowie Eingrenzung und Abwägung

A. Kafka

1) Einarbeitung in die Apache Kafka Komponente, Installation in der lokalen Entwicklungsumgebung

2) Implementierung eines ersten Producers auf Basis einer bestehenden pipe von contentCollector

3) Füllen eines Topic mit Testdaten des implementierten Producers

4) Ist die eigenene Implementierung eines Konsumenten für den nachfolgenden BatchLayer erforderlich?

B. Batch Layer auf Basis von Metafacture cluster

1) Einarbeitung in die workflows von Metafacture cluster

2) Installation in der lokalen Entwicklungsumgebung (einschliesslich Hbase und und pseudo distributed Hadoop cluster

3) Entwickeln eines ersten Datenmodells für den Storage bibliographischer Beschreibungen des swissbib service in einer 'column oriented database'.

4) Laden eines Testdatasets im Zusammenspiel mit Kafka.

25) https://github.com/dswarm/dswarm-documentation/wiki

5) Durchspielen einzelner workflows. Inwieweit ist das in der Dokumentation beschriebene Clusterung auf Basis von Metafacture möglich? Evtl. Schreiben von Unittests in einer Hadoop Umgebung.

C. Zusammenfassung der Erkenntnisse aus A. und B. und nächte Schritte

1) Erster Vergleich mit weiteren Möglichkeiten durch den Einsatz von Stream Processing Komponenten im 'Speed Layer'. Liessen sich die 2013 entwickelten Batch Verfahren in Metafacture Cluster gänzlich durch diese Verfahren ersetzen oder nur ergänzen?

2) erste konkretere Beschreibungen von neuen Anwendungen im 'Service Layer' basierend auf der neuen Architektur.

D. bewusste Eingrenzungen und Abwägungen zur Risikominimierung in der Projektarbeit

Ich habe mir eine Zeit lang überlegt, wie die Schwerpunkte in der Arbeit am besten gesetzt werden könnten. Da die Suche nach einer Alternative für unser aktuelles produktives Datenmanagement-system stark im Alltagsfokus steht und ich mir mögliche Einsatzszenarien von Metafacture_cluster schon seit einiger Zeit überlege, überwog zu Beginn der Gedanke, dies zum einzigen Arbeitspaket in der Arbeit zu deklarieren. Folgende Punkte haben mich zu einer Verbreiterung bewogen:
- Metafacture_cluster wurde bisher nur als Prototyp eingesetzt. Der damalige Entwickler hat die Deutsche Nationalbibliothek in der Zwischenzeit verlassen. Obwohl wir im Rahmen des Projekts linked-swissbib bisher gute Erfahrungen mit den Verfahren gemacht haben, kann eine ausschliess-liche Evaluation und prototypische Implementierung im Rahmen einer Projektarbeit eines CAS an zeitliche Grenzen stossen. Ausserdem gehe ich davon aus, dass die reinen batchorientierten Ver-fahren aus dem Jahre 2013 heute mit Sreamingtechnolgien zumindest ergänzt werden können. Dennoch erachte ich es immer noch als einen grossen Wert, die vor drei Jahren implementierten Verfahren einschliesslich des storage-systems als Teil unseres zukünftigen Technolgiestacks zu berücksichtigen.
- Auch durch den Besuch der diesjährigen Berlin-Buzzwords[26] ist mir deutlich geworden, wie sehr die Brokertechnolgie des Apache Kafka Projekts[27] heute zum festen Bindeglied in einer Vielzahl von Anwendungsszenarien geworden ist. Im Entwurf des Lösungsansatz zu dieser Projektskizze ist es der ideale Ersatz/Ergänzung für unsere aktuelle ingest Komponente.
- Beide Arbeitspakte zusammen bieten eine gute Basis für eine erste Bewertung eines möglichen Einsatzes von Streamingprozessen im Rahmen des 'Speed-Layers' der Lambda Architektur. Für eine erste Implementierung im Rahmen der Projektarbeit wird es jedoch nicht mehr reichen.

Im zusammengefassten Ergebnis dieser drei Pakete erhoffe ich mir eine Querschnittsanalyse, die dann wiederum als Basis für die weitere Entwicklung unserer zukünftigen Infrstruktur eingesetzt werden kann.

26) https://berlinbuzzwords.de/
27) http://kafka.apache.org/

5. Personen

Studierender	Universitätsbibliothel Basel, Projekt swissbib Günter Hipler 061 267 31 62 guenter.hipler@unibas.ch
Ansprechpartner Betreuer in der Firma:	André Golliez, Projektleiter swissbib und Präsident openData.ch http://opendata.ch/organisation/board/ andre.golliez@unibas.ch