Bereitstellung technischer Workflows für FachwissenschaftlerInnen und Übergabe an die Service-Unit (R. 2.3.4)

Version 26.02.2016
Cluster 2.3
Verantwortlicher Partner HKI

DARIAH-DE
Aufbau von Forschungsinfrastrukturen für die e-Humanities

Dieses Forschungs- und Entwicklungsprojekt wird / wurde mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF), Förderkennzeichen 01UG1110A bis N, gefördert und vom Projektträger im Deutschen Zentrum für Luft- und Raumfahrt (PT-DLR) betreut.

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

Projekt: DARIAH-DE: Aufbau von Forschungsinfrastrukturen für die e-Humanities
BMBF Förderkennzeichen: 01UG1110A bis N
Laufzeit: März 2011 bis Februar 2016

Dokumentstatus: Final
Verfügbarkeit: öffentlich

Autoren:
Johanna Puhl, HKI
Xi Kong, GWDG

Revisionsverlauf:

Datum	Autor	Kommentare
24.08.2015	Johanna Puhl	erste Version auf Englisch erstellt
06.10.2015	Johanna Puhl	Vorstellung auf F2F Treffen der AG RDLC
15.10.2015	Johanna Puhl	Deutsche Übersetzung, Ergänzung mit Diskussionsergebnissen aus der AG RDLC
08.02.2016	Xi Kong	Korrekturen und Anpassungen, insbesondere bei Workflow-Tools
12.02.2016	Xi Kong	Einarbeiten der Kommentare von Ulrich Schwardmann
26.02.2016	Xi Kong	Einarbeiten der Kommentare von Carsten Thiel

Inhalt

1. Einführung

1.1 Ziel

In diesem Dokument sollen alle technisch relevanten Standards und Spezifikationen genannt und erläutert werden, welche für die Implementation eines DARIAH-DE Research Data LifeCycle empfohlen werden.

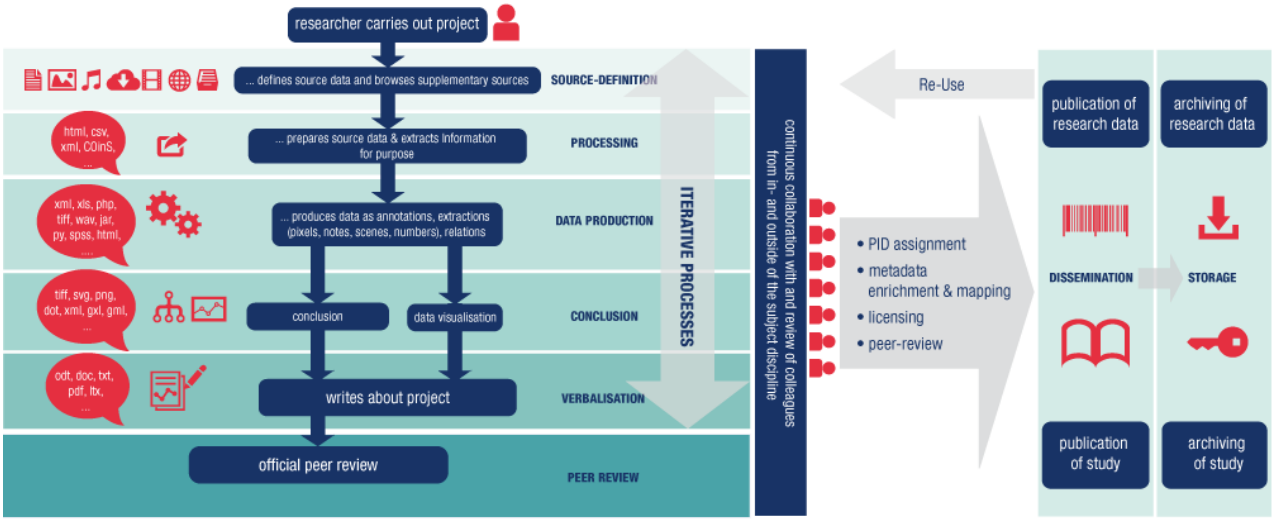
1.2 Geltungsbereich

Diese Empfehlung soll sowohl die generelle Funktionalität als auch Tools, Technologien und Abhängigkeiten enthalten, damit ein Research Data LifeCycle adäquat umgesetzt werden kann.

1.3 Referenzen

Dieses Dokument soll als technische Formalisierung die konzeptionelle Arbeit der AG Research Data LifeCycle abschließen. Bisherige, konzeptionelle Arbeiten aus der AG finden sich unter den folgenden Links:

- http://webdoc.sub.gwdg.de/pub/mon/dariah-de/dwp-2015-11.pdf
- http://www.hki.uni-koeln.de/sites/all/files/Puhl_Slides.pdf
- https://wiki.de.dariah.eu/pages/viewpage.action?pageId=38080370#Empfehlungenf%C3%BCrForschungsdaten,ToolsundMetadateninderDARIAH-DEInfrastruktur-ToolsundVerfahrenf%C3%BCrdiedigitalenGeisteswissenschaften



Externe, eher technische Referenzen werden im folgenden Abschnitt genannt. Einige Erwähnenswertere sind die Folgenden:

- Das Workflow Reference Model, welches die Basisimplementation eines Workflow management system beschreibt (http://www.wfmc.org/standards/docs/tc003v11.pdf)
- Die W3C Prov Spezifikation zur Formalisierung von Provenienz in Metadaten. (http://www.w3.org/TR/prov-overview/)
- Der WSDL Standard als Interface zu Webservices (http://www.w3.org/TR/wsdl)
- Die Taverna Software Suite (http://www.taverna.org.uk/)

2. Begriffsdefinitionen und Abkürzungen

Dieses Kapitel enthält alle notwendigen Begriffe und Abkürzungen, welche im Kontext dieses Dokuments erläutert werden müssen.

2.1 Workflows und LifeCycles

Ein Workflow kann als eine "computerbasierte Erleichterung oder Automatisierung eines Geschäftsprozesses"[1] definiert werden. Diese Definition stammt aus dem Umfeld der Wirtschaftsinformatik und bezieht sich daher naturgemäß vor allem auf wirtschaftliche Prozesse. Tatsächlich existieren Workflows natürlich darüberhinaus in allen Disziplinen und Arbeitsfeldern.
Daher empfehlen wir für die hier fokussierte Domäne der Digitalen Geisteswissenschaften den Teilbegriff "Geschäfts-" einfach weg zu lassen und optional durch den Begriff "Forschungs-" zu ersetzen. Ein Workflow ist also eine *"computerbasierte Erleichterung oder Automatisierung eines (Forschungs-) prozesses"*
Workflows können als ein **Set einzelner Komponenten beschrieben werden, aus denen sich ein Prozess zusammen setzt**. Dabei ist eine Komponente ein unteilbarer Arbeitsschritt, der (automatisiert) erledigt werden kann. Ein Workflow definiert Beziehungen zwischen diesen Komponenten, d.h. in Form einer **Informations- oder Datenübertragung von einer Komponente zu einer anderen**. In einem Workflow Management System soll ein Workflow durch die Instantiierung des dahinter liegenden Workflow Datenmodells bereit gestellt werden.
Häufig werden die Konzepte Workflow und LifeCycle verwechselt oder synonym verwendet. Um zwischen die beiden Begriffen Klarheit herzustellen, soll Folgendes gelten[2]:

Lifecycles werden implementiert, um den (lifecycle-) Status von Objekten und Dateien in einem System zu kontrollieren. Ein LifeCycle Status für ein Objekt kann beispielsweise "created" oder "draft" zu Beginn und "completed" oder "published" am Ende eines

[1] Deutsche Übersetzung von WfMC, Glossary, Terminology and Glossary, 3rd Edition. Document No WFMC-TC-1011. Workflow Management Coalition. Winchester, 1999
[2] Vgl. http://iieblogs.org/2012/04/18/lifecycles-versus-workflows-separate-and-together-build-your-pdm-systems/

LifeCycles sein. Während eines LifeCycle könnte ein Objekt sich entsprechend “in review” or “in refinement” befinden. Durch die Etablierung eines LifeCycle Models kann klarer definiert werden, welchen Status ein Objekt oder eine Datei erreichen kann.
Das Management eines Lifecycles beinhaltet außerdem die Verwaltung von Nutzern und damit verbundenen Rollen und Rechten. So kann kontrolliert werden, wer in welchem Status mit welchem Objekt interagieren darf.
Folgende Definition soll dabei gelten: Workflows regeln alle Implikationen, die mit jedem LifeCycle-Status eine Objekt verbunden sind. Ein Workflow Management Tool verarbeitet **Aufgaben innerhalb eines LifeCycle-Status und zwischen** LifeCycles. Zum Beispiel kann ein Workflow Prozesse außerhalb eines LifeCycles verwalten, wenn sie durch eine LifeCycle Statusänderung angesteuert werden. Workflow Management Tools können zudem Workflows einzelnen Nutzern zuweisen oder Aufgaben von einem zum anderen Nutzer delegieren.
Es lässt sich also folgende Beziehung definieren:
LifeCycles sind die theoretische Basis zur Implementation von konkreten Workflows, welche wiederum eine Kette von Status als Instanzen eines abstrakten LifeCycles definieren.

2.2 Service

Ein Service wird definiert als ein “set of related software functionalities that can be accessed via application programming interface (API).“[3] Allgemein besteht die Anforderung, dass ein Service REST-ful und im Kontext von Workflow Management möglichst auf der gleichen Granularitätsebene wie eine Workflow-Komponente (nämlich als Instanz einer Workflow Komponente) eingesetzt werden soll.

2.3 Tools

Der Tool-Begriff bezieht sich in der bisherigen theoretischen Diskussion geisteswissenschaftlicher Arbeit auf die praktische Ausführung einer bestimmten wissenschaftlichen Aktivität durch den Einsatz einer Software. Im Kontext der technischen Implementation eines Frameworks kann ein Tool aber darüberhinaus die Ausführung von administrativen oder anderen nicht in der direkten wissenschaftlichen Fragestellung begründeten Aufgaben abdecken.
Der Aufruf eines Tools sollte innerhalb eines Workflows – möglichst definiert durch eine Datei, welche vom Workflow Datenmodell abgeleitet ist – erfolgen.
Ein Tool wird innerhalb einer solchen Konstruktion als Service aufgerufen.

2.4 Objekte vs. Dateien

Eine Datei ist ein definierter und in sich konsistenter Strom von binären Signalen (Bitstream) während Objekte eher unscharfen Definitionen unterliegen und daher eine Definition festgelegt werden muss.

[3] Gemäß milestone 1.3.4.1 in DARIAH I: "Service Interoperability – Multi-Modal Interoperability Concept". S.4

Wir halten uns daher an die Objektdefinition des Nestor Netzwerks[4], das zwischen physischen, logischen und konzeptionellen Objekten unterscheidet.
Dieses Dokument beschäftigt sich hauptsächlich mit Objekten der physischen und logischen Ebene, also Dateien. Der konzeptionelle Kontext kann nur mithilfe von deskriptiven Metadaten modelliert werden (siehe nächster Abschnitt).

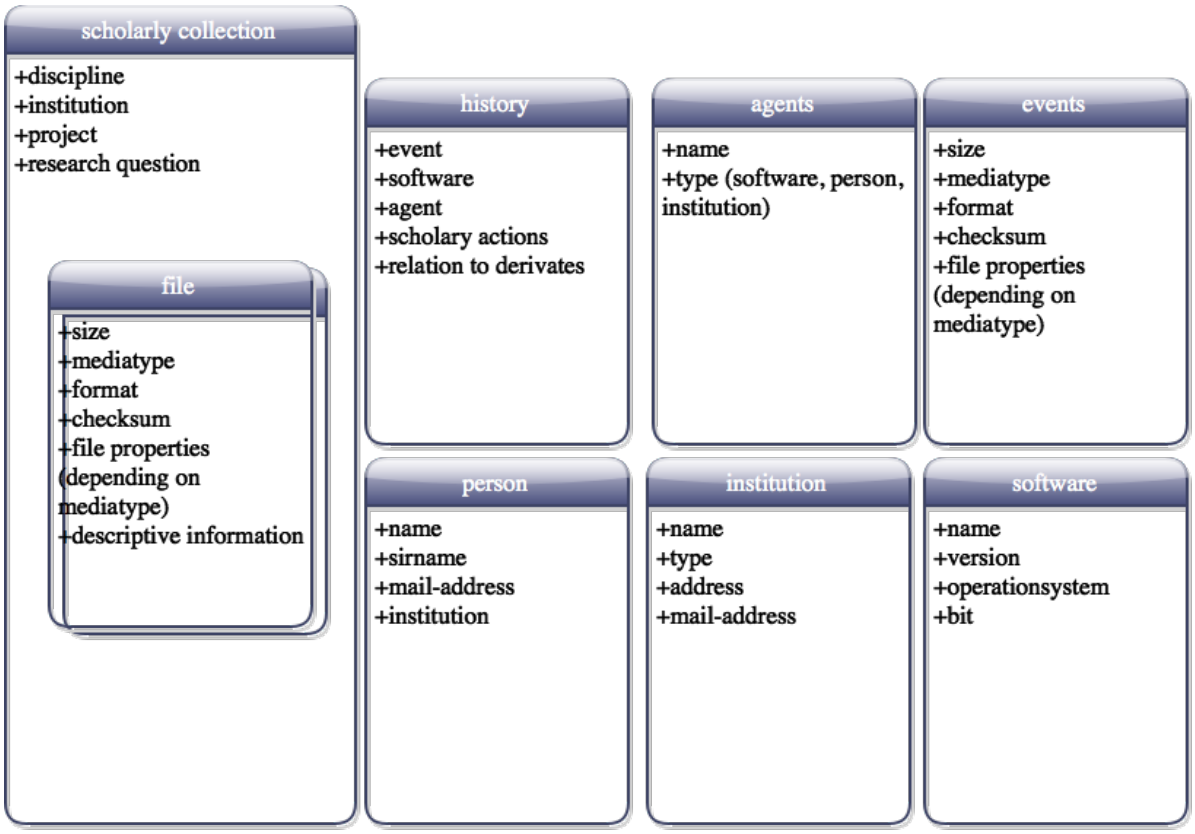
2.5 Metadaten

Metadaten, als Daten (Informationen) über Daten unterstützen ein System und deren Benutzung auf unterschiedliche Weise. Daher differenzieren wir zwischen unterschiedlichen Arten von Metadaten: administrative, strukturelle und deskriptive Metadaten.[5] Das beschriebene Framework soll alle drei Arten unterstützen aber zur Vermeidung von Redundanz diese möglichst scharf getrennt voneinander gemäß den unterschiedlichen Standards vorhalten.

- Deskriptive Information muss in **DublinCore** abgelegt werden und kann gegebenenfalls darüberhinaus in weiteren Standards abgelegt werden
- Die Struktur von Forschungsdaten und Forschungsdatensammlungen soll mit dem **Dariah Collection Description Data Model (DCDDM)**[6] abgedeckt werden
- Administrative und Provinienz-Information sollten in **W3C PROV**[7] und mindestens marginal in **DublinCore** gespeichert werden.
- Dateiinformationen sollen mithilfe der Software **JHOVE** oder mithilfe des Softwarepakets **fido extrahiert und in W3C Prov** gespeichert werden.
- Der scholastische Kontext, also die scholastische Disziplin, verwendete Tools und Methoden sollten auch in einem entsprechenden Standard modelliert werden – hier besteht noch eindeutig Entwicklungsbedarf, ggf. bietet sich das **SDM** Modell o.ä. an.

Die folgende Abbildung gibt (keine vollständige) Übersicht über erforderliche Metadatentypen und -felder:

[4] nestor 2010, Kap 9.1, S. 4
[5] http://www.loc.gov/standards/metadata.html
[6] Dariah Collection Description Data Model (https://github.com/DARIAH-DE/DCDDM)
[7] Vgl. http://www.joanneum.at/uploads/tx_publicationlibrary/BAW-ipres2014_mpaf_cr_v4.pdf



Konzept eines exemplarischen Metadaten-Models für Forschungsdaten

2.6 Forschungsdaten

Folgende Definition von Forschungsdaten gilt in DARIAH-DE:

> *"Unter digitalen geistes- und kulturwissenschaftlichen Forschungsdaten werden innerhalb von DARIAH-DE all jene Quellen und Ergebnisse verstanden, die im Kontext einer geistes- und kulturwissenschaftlichen Forschungsfrage gesammelt, beschrieben, ausgewertet und/oder erzeugt wurden und in maschinenlesbarer Form zum Zwecke der Archivierung, Zitierbarkeit und zur weiteren Verarbeitung aufbewahrt werden können."*[8]

2.7 Provenienz

Der Provenienzbegriff wird hier technisch verstanden. Die Provenienz, also Herkunft und Geschichte eines Objekts vor dem Zeitpunkt der Digitalisierung, welche für Geisteswissenschaftler hochinteressant sein können, kann nicht vollumfänglich berücksichtigt werden.
Wenn solche Informationen aber als deskriptive Metadaten durch Fachwissenschaftler manuell bereit gestellt werden und entsprechend gekennzeichnet sind, kann über ein

[8] Johanna Puhl, Peter Andorfer, Mareike Höckendorff, Stefan Schmunk, Juliane Stiller, Klaus Thoden: "Diskussion und Definition eines Research Data LifeCycle für die digitalen Geisteswissenschaften". DARIAH-DE Working Papers Nr. 11. Göttingen: DARIAH-DE, 2015 URN: urn:nbn:de:gbv:7-dariah-2015-4-4

Mapping die Möglichkeit bereit gestellt werden, die Informationen in geeignete Felder eines Provenienz-Metadatenstandards zu überführen.

3. Funktionen und funktionale Anmerkungen

Neben forschungsrelevanten, also inhaltlichen Tools und Diensten wird der basale Forschungsdatenzyklus in dem hier besprochenen Framework durch die folgenden Funktionen unterstützt, welche jeweils von einer Workflowkomponente abgedeckt werden:

3.1 Basiskomponenten

Die folgenden Komponenten definieren den Basisworkflow für einen wissenschaftlichen Data LifeCycle sind aber nicht hinreichend spezifisch um als Workflow, der sich dezidiert an die Geistes- und Sozialwissenschaften richtet, gelten zu können.
Diese Basiskomponenten sollen als Rest-ful APIs eingebunden werden:

- Ingest Service (DARIAH Publish Gui 2.0)
- Generierung einer W3C Prov Datei im Ingest
- Sammlungsdefintion im Ingest über das DCDDM Profil https://github.com/DARIAH-DE/DCDDM
- PID-Service (Epic Api)
- Nutzerverwaltung (Add, Edit, Delete) Function (AAI -LDAP)
- Review Service
- Annotation Service (SemToNotes oder annotator.js)
- Kollaboration Service (Etherpad API, Wiki API)
- Publikation Service (DARIAH Publish Gui 2.0, DARIAH-DE Repository)
- Bitstream Preservation Service (DARIAH Bitstream Preservation)

Darüberhinaus benötigt man zur Definition, Bearbeitung und Löschung von Workflows sowie zu deren automatischer Ausführung entsprechende Funktionen in einem Management System. Minimal also:

- Eine Workflow Definitions Funktion (Add, Edit, Delete)
- Eine Workflow Execution Funktion / Engine
- Ein Datenmodell mithilfe dessen man diese Workflows beschreiben und wiederholen kann.

Die letzten 3 Punkte zur Implementierung eines Workflow Management sollen durch ein entsprechendes Softwarepaket (Taverna Workbench 2.5) abgedeckt werden.

3.2 Weitere notwendige und fortgeschrittene Funktionen

Wie in Abschnitt 3.1 schon erwähnt, sollte neben den genannten Basiskomponenten außerdem eine gewisse Grundmenge an inhaltsbezogenen disziplinrelevanten Funktionen angeboten werden. Hier sind Tools zur Durchführung von quantitativen Analysen,

konfigurierbare Mustererkennung für verschiedene Medientypen, Visualisierungstools, die (teil-) automatische Integration von kontrollierten Vokabularen und Normdaten u.v.m. denkbar. [9]
Wenn diese Funktionen entsprechend im Framework unterstützt werden, können sie in speziell auf geisteswissenschaftliche Fragestellungen ausgerichteten Workflows eingesetzt werden.
Um externe Tools in Workflows als Komponenten integrieren zu können, müssen diese über Schnittstellen, wie WSDL, resp. WADL angesprochen werden. Wenn die Gefahr besteht, dass zu große Datenmengen übers Netz übertragen werden, kann es auch sinnvoll sein, ein Tool lokal auf einem Rechner nur zur Durchführung eines bestimmten Forschungssettings zu installieren und entsprechend lokal anzusteuern.
Die folgenden Anforderungen sollen für alle Komponenten gelten:

3.3 Notwenige technische Anforderungen

Das angestrebte Framework wird durch die **Taverna Software Suite**[10] umgesetzt, welche entsprechend den Wünschen von Geisteswissenschaftlern mit vorkonfigurierten Workflows angepasst werden soll.
Für die Umsetzung des Frameworks in einer solchen Software Suite sollen unabhängig von konkreten Distributionen folgende Anforderungen gelten:

- Das angestrebte Software System soll unter **JAVA** entwickelt sein, damit es sowohl plattformunabhängig läuft als auch als Server-Client Architektur betreibbar ist.
- Das angestrebte Software System sollte modular aufgebaut sein, so dass sich einzelne Komponenten mit begrenztem Aufwand austauschen lassen oder das Gesamtsystem erweiterbar bleibt.
- Es wird ein vollständiges, gut dokumentiertes Open Source System bevorzugt.
- Die voll-automatische Ausführung eines Workflows sollte nicht mehr Zeit als **1 Stunde** beanspruchen
- Im Falle von eingeplanten manuellen Eingriffen (User-Review etc. als ask Funktion) in Workflows sind auch Ausführungszeiten von **1 Woche** akzeptabel.
- **Wiederholungen** eines Workflows sind zur Kontrolle und Nachvollziehbarkeit in beiden Fällen sinnvoll und ausdrücklich erwünscht.
- Die Größe einzelner Dateien sollte die Grenze von 2 GB alleine wegen der Begrenzung des http Protokolls nicht überschreiten. Eine deutlich kleinere Dateigröße (unter 100 MB pro Datei) beschleunigt die meisten Vorgänge aber wesentlich.
- Die Architektur sollte alle gängigen Schnittstellen, mindestens aber Rest / WSDL / WADL und lokale Anbindungen unterstützen
- Das System sollte sowohl serverseitig als auch clientseitig steuerbar sein

[9] Eine Empfehlung für entsprechende Tools kann hier abgerufen werden: https://dev2.dariah.eu/wiki/pages/viewpage.action?pageId=38080370#EmpfehlungenfürForschungsdaten,ToolsundMetadateninderDARIAH-DEInfrastruktur-ToolsundVerfahrenfürdiedigitalenGeisteswissenschaften.
[10] http://www.taverna.org.uk/ Hier am Besten die Taverna Workbench oder der Taverna Server in der Version 2.5 für Digital Preservation

3.4 Auswirkungen auf und Einflüsse durch Nutzer

3.4.1 Nutzerverwaltung, Rollen- und Rechte

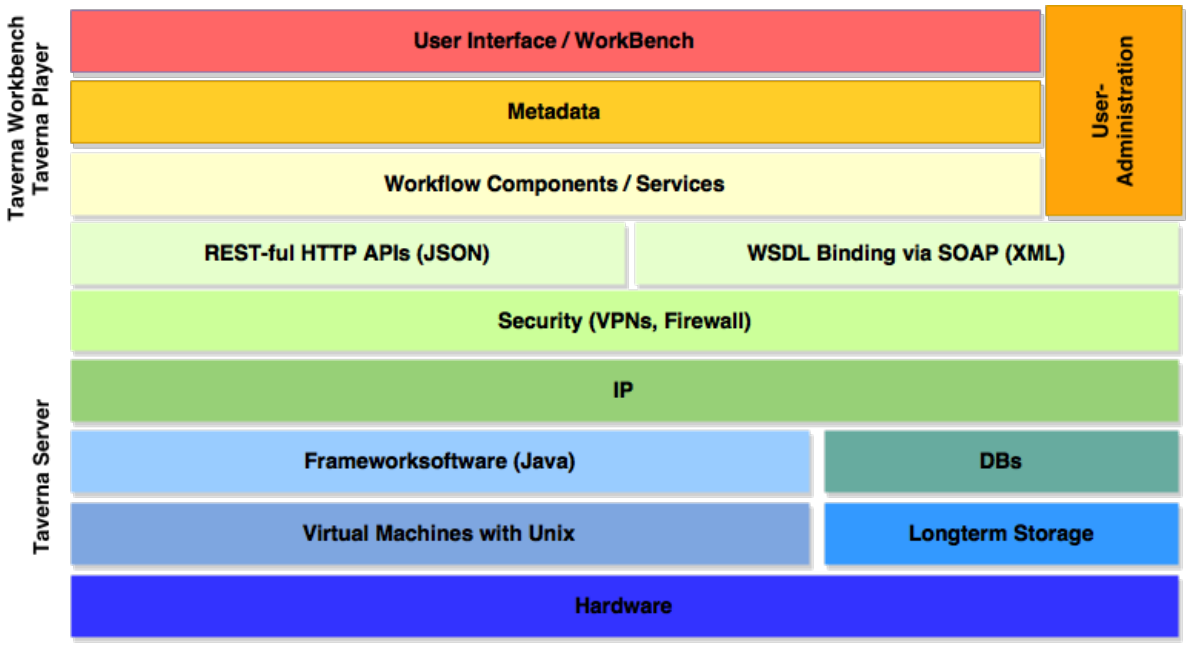
Sowohl für die Durchführung als auch für die anschließende Begutachtung und Publikation von Forschungsdaten ist die Implementation einer Nutzerverwaltung und eines Rollen- und Rechtemodells unausweichlich.
Dabei soll das Basismodul zur Nutzerverwaltung durch das technische System bereitgestellt werden. Hier kann das LDAP Protokoll empfohlen werden, speziell die AAI Schnittstelle im DFN LDAP wird bereits in DARIAH-DE unterstützt.
Diese Schnittstelle muss allerdings für jeden Workflow im Workflowsystem neu konfiguriert werden, da entsprechende Zugriffsrechte auf Dateien und Ausführungsrechte für einzelne Komponenten (Tools) angepasst werden müssen. Ebenso muss das hinter der AAI liegende Rollen- und Rechtemodell daraufhin überprüft werden ob es für den Einsatz in einem entsprechend feingranular genutzten Workflow-Managmentsystem geeignet ist.
Angebunden an die Verwaltung von Rollen- und Rechten sollte ein Modul zur Vergabe einheitlicher Lizenzvereinbarungen zur Nutzung und Veränderung der Forschungsdaten in das Workflow-Managementsystem aufgenommen werden.

3.4.2 UseCases

Es ist unbedingt notwendig, das geplante Workflow-Managementsystem vorab mit geisteswissenschaftlichen UseCases aus möglichst unterschiedlichen Teildisziplinen zu testen, damit eine entsprechende Eignung sicher gestellt wird.
Das Kapitel zu Empfehlungen und dem weiteren Vorgehen geht genauer auf entsprechende Pläne ein.

3.5 Visualisierung des Technology Stacks

Die folgende Abbildung ist der Versuch, alle notwendigen und voneinander abhängigen Technologien darzustellen und diese direkt in Beziehung zum geplanten Einsatz der Taverna Software Suite zu setzen.



Technology Stack für ein LifeCycle Framework

4. Empfehlung und weiteres Vorgehen

4.1 Software

Zur Umsetzung des LifeCycle-Frameworks soll die Software Taverna (Workbench, bzw. Server) in der Version 2.5 Beta getestet und dann verwendet werden. Sollte in einem Zeitraum von bspw. 6 Monaten die Version 3 Alpha in einem Zustand sein, dass hier auch das Digital Preservation Modul funktionsfähig ist, kann rechtzeitig ein entsprechender Versionswechsel statt finden.
Ziel ist die Bereitstellung einer Taverna Workbench for Digital Humanities mit vorkonfigurierten Workflows, Tools und Standards aus DARIAH.

Aufgrund der aktuellen Evaluation wird hier zwar Taverna als Workflow Tool für Digital Humanities empfohlen, es werden allerdings, unter Vorbehalt der Betriebstauglichkeit, andere leistungsfähigere Workflow Management Tools nicht ausgeschlossen. Außerdem sollten hierbei dringend einige Anwendungen aus der Digital Humanities Domain mit der Software geprüft und getestet werden.

4.2 Provinienz

Taverna 2.5 wird hinsichtlich der schon bestehenden Integration eines Provenienz Metadatenstandards überprüft. Hier kann bereits auf ein Plugin zur Unterstützung eines

entsprechenden Standards zurück gegriffen werden (W3C Prov).[11] Zusätzlich kann speziell für Anforderungen an die digitale Langzeitarchivierung auch ein Mapping nach PREMIS statt finden, welches dann in einer eigenen Workflow Komponente implementiert werden soll.

4.3 UseCases

Eine Bewährungsprobe ist der erfolgreiche Test der Taverna Workbench mit den in DARIAH-DE entwickelten UseCases.
Hier wurden in DARIAH-DE folgende UseCases[12] erarbeitet:

- Narrative Techniken und Untergattungen im deutschen Roman
- Biografien
- Identifikation von griechischen und lateinischen Texten in einer Sammlung von 2 Mio. Texten

Für die Anpassung als DARIAH Taverna Workbench gilt es entsprechend die Taverna Workbench mit den benötigten APIs zu bestücken und die entsprechenden Corpora einzubinden.

4.4 Klassifikation von Workflows, Anbindung an den SLC

Eine Verbindung zu den Arbeiten der AG Service Life Cycle besteht vor allem in der Dokumentation von Services, so dass sie entsprechend über APIs in einem Managementsystem abgefragt werden können. Dabei kann beispielsweise eine entsprechende Datenbank (analog zu http://www.myexperiment.org/) die Recherche nach geeigneten Services vereinfachen.
Daneben wird eine Klassifikation von DARIAH Workflows und Services gemäß einem entsprechenden kontrollierten Vokabular, wie dem Scholarly Domain Model / NeDimah / TaDirah o.ä. empfohlen, um einen besseren Zugang und eine bessere Übersicht über die Möglichkeiten für geisteswissenschaftliche Forschungsfragen zu ermöglichen.

4.5 Rechtliches

Zur rechtlichen Dimension gilt, dass alle geschriebene Software sowie alle verwendete Software dem Open Source Paradigma unterliegen und mit entsprechenden Lizenzen gekennzeichnet werden soll.
Aus diesem Grund wird hier eine Empfehlung für Apache 2.0 ausgesprochen, da diese bereits für die Taverna Software Suite verwendet wird.

[11] Vgl. https://github.com/taverna/taverna-prov
[12] Vgl. https://wiki.de.dariah.eu/download/attachments/26150061/R%205.2.1%20%E2%80%93%20Beschreibung%20der%20Use%20Cases.pdf?version=1&modificationDate=1424423861383&api=v2