DARIAH-DE Repositorium – Prototyp (M 4.3.2.1)

Version 27. April 2015

Cluster 4

Verantwortlicher Partner SUB Göttingen

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: public

Autoren:

Stefan E. Funk, SUB Göttingen

Stefan Schmunk, SUB Göttingen

Revisionsverlauf:

Datum	Autor	Kommentare
22.04.2015	Stefan E. Funk	Erste Version
24.04.2015	Stefan Schmunk	Einleitung ergänzt und allgemeine Überarbeitung
24.04.2015	Stefan E. Funk	Anleitung zur Publish GUI, Literaturverzeichnis und Abbildungsverzeichnis ergänzt
27.04.2015	Stefan E. Funk	Anleitung zur Publish GUI fertig gestellt
27.04.2015	Stefan Schmunk, Stefan E. Funk	Endredaktion

Inhalt

1. Einleitung

Im Rahmen von DARIAH-DE widmet sich das Cluster „Wissenschaftliche Sammlungen und Forschungsdaten" nicht nur methodischen und konzeptionellen Fragen des Umgangs, der Generierung, der Nutzung[1] und des Enrichments von digitalen Forschungsdaten, sondern ein zentraler Teil der Tätigkeiten besteht in der Entwicklung und Realisierung einer Repository-Lösung für geistes- und kulturwissenschaftliche Forschungsdaten.[2]

Das DARIAH-DE Repositorium wird zukünftig nicht nur DARIAH-DE assoziierten Forschungsprojekten zur Verfügung stehen, wie derzeit beispielsweise TextGrid[3], sondern auch Einzelforscherinnen und -Forschern sowie Forschungsprojekten, die ihre Forschungsdaten persistent, referenzierbar und langzeitarchiviert speichern und Dritten zur Verfügung stellen wollen. Ebenfalls sind Wissenschaftlerinnen und Wissenschaftler an Universitäten und Forschungseinrichtungen adressiert, die in Forschungsprojekten entstandene, erhobene, erfasste und/oder generierte Forschungsdaten langfristig im Rahmen einer Repository-Lösung speichern wollen. Hierbei steht vor allem der einfache nutzerorientierte (Usability) versehene Zugang von Fachwissenschaftlerinnen und Fachwissenschaftlern zu einer Langzeitspeicherung von Forschungsdaten im Vordergrund. Das DARIAH-DE Repositorium ermöglicht es, Forschungsdaten zu speichern, mit Metadaten zu versehen, diese durch die Generische Suche aufzufinden und vor allem durch die Nutzung von EPIC-PIDs[4] eine permanente (maschinenlesbare) Referenzierung zu ermöglichen.

Um dies zu erreichen, arbeiten DARIAH-DE und TextGrid, das aus der Virtuellen Forschungsumgebung TextGrid Laboratory (TextGridLab)[5] und dem TextGrid Repository (TextGridRep)[6] besteht, zusammen. Das DARIAH-DE Repositorium stützt sich auf die Codebasis des TextGrid Repository und wurde mit verschiedenen Service-Instanzen und unterschiedlichen an das DARIAH-DE Repositorium angepassten Modulen für Funktionen wie Speicher- und AAI-Zugriff implementiert.

Im Projekt DARIAH-DE wurde in den vergangenen Jahren u. a. eine Authentifizierungs-und Autorisierungsinfrastruktur (AAI) und die DARIAH-DE Storage API für

---

[1] Aber auch Nutzungsmöglichkeiten wie z. B. lizenzrechtlichen Fragen, siehe: https://de.dariah.eu/lizenzen

[2] Vgl. Forschungsdaten in DARIAH-DE. https://de.dariah.eu/forschungsdaten

[3] Vgl. TextGrid: Digital edieren – forschen – archivieren. http://textgrid.de/

[4] Vgl. Nachhaltige Referenzierung von Digitalen Objekten mit Hilfe von persistenten Identifikatoren (PID). https://de.dariah.eu/pid-service

[5] Vgl. TextGrid – Download und Installation. https://www.textgrid.de/registrierungdownload/download-und-installation/

[6] Vgl. TextGrid Repository. http://www.textgridrep.de/

die Speicherung von Forschungsdaten auf Bit Preservation Level aufgebaut, so dass Forschungsdaten zwischen den beteiligten Rechenzentren repliziert werden. Dadurch ist sichergestellt, dass die Forschungsinfrastruktur nicht nur als Speicherort für statische Daten verwendet werden kann – diese also öffentlich zugänglich, zitierfähig und langzeitarchiviert sind – sondern ebenso die Möglichkeit gegeben ist, dynamische Daten – die gegebenenfalls durch eine AAI gesichert sind und die aufgrund andauernder aktiver Nutzung aktualisiert werden müssen – dort abzulegen.

Auf die Forschungsdaten kann mithilfe von APIs (maschinenlesbar) zugegriffen werden und zugleich werden alle Forschungsdaten mit EPIC-PIDs versehen, so dass andere Tools und Service diese nachnutzen können.[7] Zu diesen Tools gehört beispielsweise die DARIAH-DE Collection Registry[8]. Sie enthält Informationen über beliebige Forschungsdaten-Repositorien und deren Sammlungen. Die in DARIAH-DE entwickelte Generische Suche[9] indiziert die Sammlungen der Collection Registry und bietet so einen userfreundlichen und zudem konfigurierbaren Zugriff auf die Inhalte. Die dritte Komponente bildet die DARIAH-DE Schema Registry, die eng mit der Generischen Suche vernetzt ist und das Mapping unterschiedlichster Metadatenbeschreibungen von Sammlungen ermöglicht. Diese stellt die XML-Schemata für das Mapping und für Metadata Crosswalks zur Verfügung. (siehe Abbildung 1: Dienste von DARIAH-DE und das TextGrid/DARIAH-DE Repositorium).

[7] Eine Übersicht der verzahnten Applikationen, die zur Speicherung, zur Suche und Recherche und den Zugang zu Forschungsdaten ermöglichen, findet sich hier: https://de.dariah.eu/forschungsdatensammlungen

[8] Vgl. DARIAH-DE – Collection Registry. https://de.dariah.eu/collection-registry

[9] Vgl. DARIAH-DE – Generische Suche. https://de.dariah.eu/generische-suche

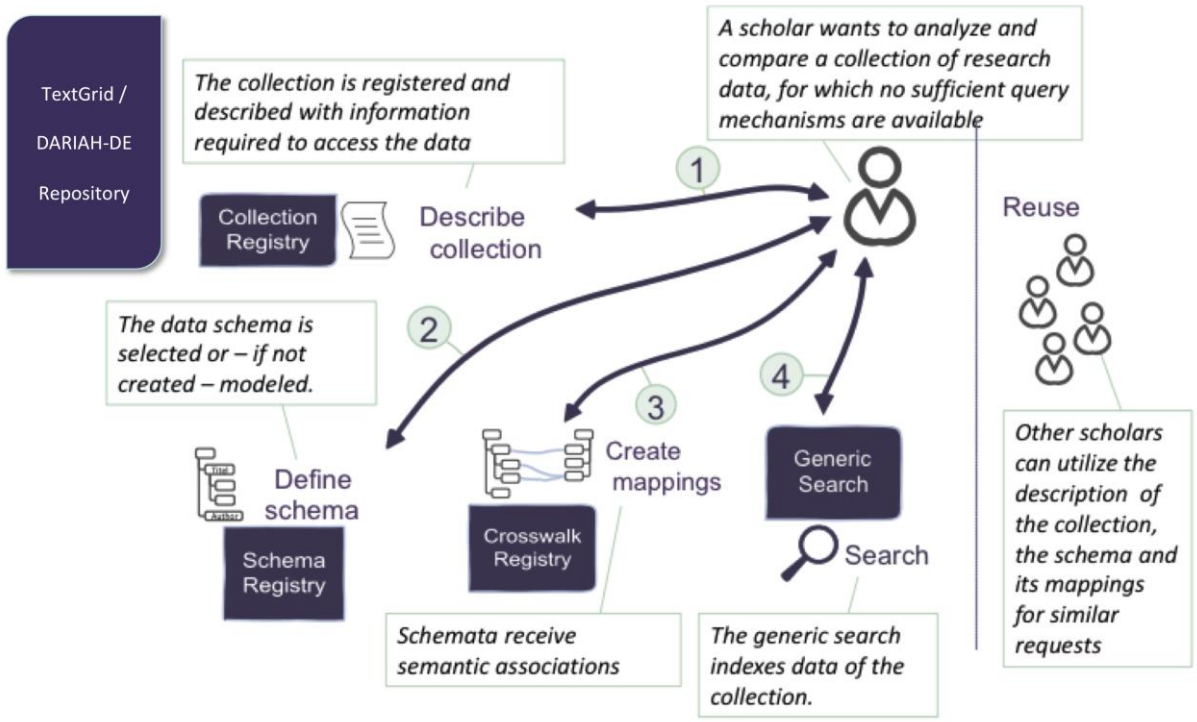

Abbildung 1: Dienste von DARIAH-DE und das TextGrid/DARIAH-DE Repositorium

Wie die Abbildung zeigt, wurde bewusst der Fokus auf einen modularen Aufbau der DARIAH-DE Forschungsinfrastruktur gelegt. Alle vorhandenen Tools und Services sind auch einzeln z. B. in anderen Projektkontexten nutzbar, die Implementierung und der Betrieb weiterer Instanzen durch Dritte ist ebenfalls jederzeit möglich. Zugleich – und dies ist die essentielle Stärke dieses architektonischen Ansatzes –, können die über die Generische Suche such- und findbaren Sammlungsbeschreibungen und Forschungsdaten auch aus anderen Registries bzw. Repositorien stammen. Dieser föderative Ansatz bietet so die Möglichkeit, sowohl die einzelnen von DARIAH-DE entwickelten und betriebenen Komponenten als „Gesamtsystem“ zu nutzen, aber zugleich können Projekte und Einrichtungen über definierte APIs die Nachnutzung ihrer eigenen Sammlungsbeschreibungen und Forschungsdaten ermöglichen (siehe Abbildung 2: Das DARIAH-DE Repositorium und angeschlossene Dienste).

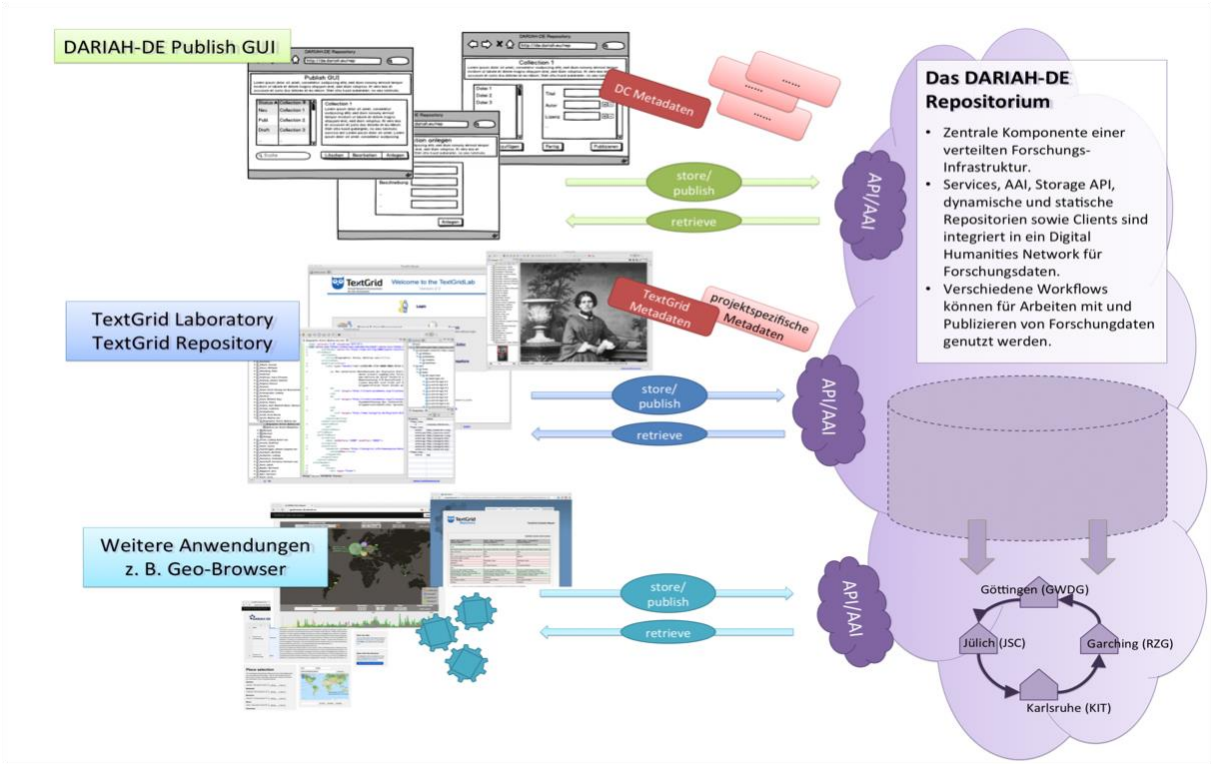

Abbildung 2: Das DARIAH-DE Repositorium und angeschlossene Dienste

Im Rahmen dieses Reports werden die Architektur und die verwendeten Technologien des DARIAH-DE Repositoriums beschrieben. Darüber hinaus wird auf den Publikationsprozess von Forschungsdaten und des der Sammlungsbeschreibung zugrunde liegenden Datenmodells eingegangen. Neben den technischen und administrativen Aspekten wird exemplarisch der Vorgang des Dateningest in das DARIAH-DE Repositorium und der Registrierung von Sammlungsbeschreibungen in der Collection Registry beschrieben, für deren Umsetzung prototypische Arbeits- und Forschungsprozesse bereits während des Entwicklungsprozesses herangezogen wurden. Die gesamten Entwicklungstätigkeiten erfolgten in enger Abstimmung mit den Fachwissenschaftlerinnen und Fachwissenschaftlern des Konsortiums und werden in den kommenden Monaten auch auf dieser Basis weiter vorangetrieben.

An dieser Stelle soll kurz auf die weiteren Entwicklungstätigkeiten bis Projektlaufzeitende eingegangen werden. Während der Prototyp des DARIAH-DE Repositoriums vor allem auf den manuellen Ingest von Forschungsdaten via eines Webinterfaces und der Collection Registry ausgerichtet ist, soll insbesondere durch die weiteren Entwicklungstätigkeiten die Möglichkeit eines „Masseningest“ von Forschungsdaten mithilfe der bereits entwickelten APIs ermöglicht werden. Abbildung 2 zeigt darüber hinaus aber noch eine weitere Anwendungsmöglichkeit. Forschungsdaten werden in den Digital Humanities oftmals durch digitale Werkzeuge und Services erstellt, erzeugt bzw. analysiert. Zwar können die originären Daten z. B. aus bereits bestehenden digitalen Sammlungen, Bibliotheksbeständen und/oder Archiven stammen, aber diese stehen oftmals am Beginn des Forschungsprozesses und werden durch digitale Verfahren aggregiert, angereichert,

und/oder annotiert, aber vor allem präprozessiert, so dass diese selbst zu einer neuen Sammlung werden. Aus diesem Grund soll in den kommenden Monaten erprobt werden, ob und wie ein Ingest von Forschungsdaten aus einzelnen Anwendungen direkt in das DARIAH-DE Repository möglich ist. Auf diese Weise sollen Wissenschaftlerinnen und Wissenschaftlern z. B. nach einer (Prä)-Prozessierung, eines Enrichments und/oder einer Analyse von Forschungsdaten in einem Tool die Möglichkeit erhalten, ihre gewonnenen Forschungsdaten unmittelbar aus der Anwendung heraus in das DARIAH-DE Repositorium zu publizieren. Dies ist die Voraussetzung, dass durch digitale Methoden und Tools gewonnene bzw. generierte Forschungsdaten zur Validierung bzw. Falsifizierung von Forschungsergebnissen herangezogen und referenziert werden können. Die Möglichkeiten sollen anhand des DARIAH-DE Geo-Browsers[10] exemplarisch getestet werden.

Der Report baut sich aus vier inhaltlichen Kapiteln auf. Im ersten Kapitel wird der Publikationsvorgang des DARIAH-DE Repositoriums erläutert und insbesondere das Zusammenspiel der Generischen Suche, der Schema Registry, der Collection Registry mit dem DARIAH-DE Repositorium beschrieben werden. Hierbei liegt der Fokus insbesondere auf dem modularen Zusammenspiel der einzelnen Komponenten. In einem zweiten Kapitel werden die einzelnen für die Publikation notwendigen Schritte aus Anwendersicht beschrieben und dadurch exemplarisch die Nutzungsmöglichkeiten aufgezeigt. Während im dritten Kapitel vor allem weitere Entwicklungsmöglichkeiten skizziert werden, dient das abschließende vierte Kapitel, um die verwendeten technologischen Komponenten ausführlicher zu beschreiben.

[10] Vgl. DARIAH-DE – Geo.Browser. https://de.dariah.eu/geobrowser

2. DARIAH-DE Repositorium – Publikationsvorgang

Das DARIAH-DE-Repositorium ermöglicht es DARIAH-Nutzerinnen und -Nutzern, ihre digitalen Objekte bzw. Datensammlungen/Kollektionen nachhaltig und sicher zu archivieren. Dies ist komfortabel und intuitiv über ein Web-Interface des DARIAH-DE-Portals im Browser möglich. Hierbei handelt es sich um die DARIAH Publish GUI. Eine Kollektion wird zunächst vom Nutzer über die Publish GUI angelegt und mit Metadaten ausgezeichnet. Einer Kollektion können beliebige Dateien zugeordnet werden, die über die GUI hochgeladen und ebenfalls mit Metadaten ausgezeichnet werden können. Die Daten werden unmittelbar nach dem Publizieren per Persistent Identifier (PID) referenziert, damit öffentlich zugänglich, und die Kollektion wird in der DARIAH-DE Collection Registry nachgewiesen. Sobald die Kollektion selbst über die Collection Registry publiziert wurde, sind die Daten mit der Generischen Suche von DARIAH-DE recherchierbar.

Für das DARIAH-DE-Repositorium sind einige Services nötig, die in Teilen von bereits existierenden Diensten des Projekts TextGrid stammen, angepasst wurden und somit nachgenutzt werden können. Für eine komfortable Nutzung des DARIAH-DE-Repositoriums ist unter Anderem das Konzept der Kollektion dringend nötig, so dass z. B. die Beziehungen zwischen digitalen Objekten innerhalb einer solchen Sammlung abgebildet werden können (als digitales Objekt wird hier eine Datei samt ihrer zugehörigen beschreibenden Metadaten verstanden). Weiterhin kann auf eine Metadaten-Eingabe und gerade auch auf eine Metadaten-Validierung nicht verzichtet werden, so dass ein einfacher CRUD-Service – mit nur den Funktionen CREATE, RETRIEVE, UPDATE und DELETE – nicht als ausreichend erachtet wurde.

Der Workflow für einen Import in das Repository wird im Folgenden kurz dargestellt (Abbildung 3: Architektur des DARIAH-DE Repositorium Prototyps). Die Authentifizierung erfolgt über die DARIAH AAI[11] und muss von allen Services bedient werden.[12] Auf diese Weise können alle Nutzer, die über einen DARIAH-DE Account verfügen und für die Nutzung des DARIAH-DE Repositoriums freigeschaltet wurden, das Repositorium nutzen und Daten speichern.

[11] Vgl. DARIAH Authorization and Authentication Infrastructure. https://dev2.dariah.eu/wiki/download/attachments/6783645/DARIAH-AAI-Concept-v0.3a.pdf?version=1&modificationDate=1328883126670&api=v2

[12] Vgl. hierzu auch Kapitel 5: Technische Grundlagen.

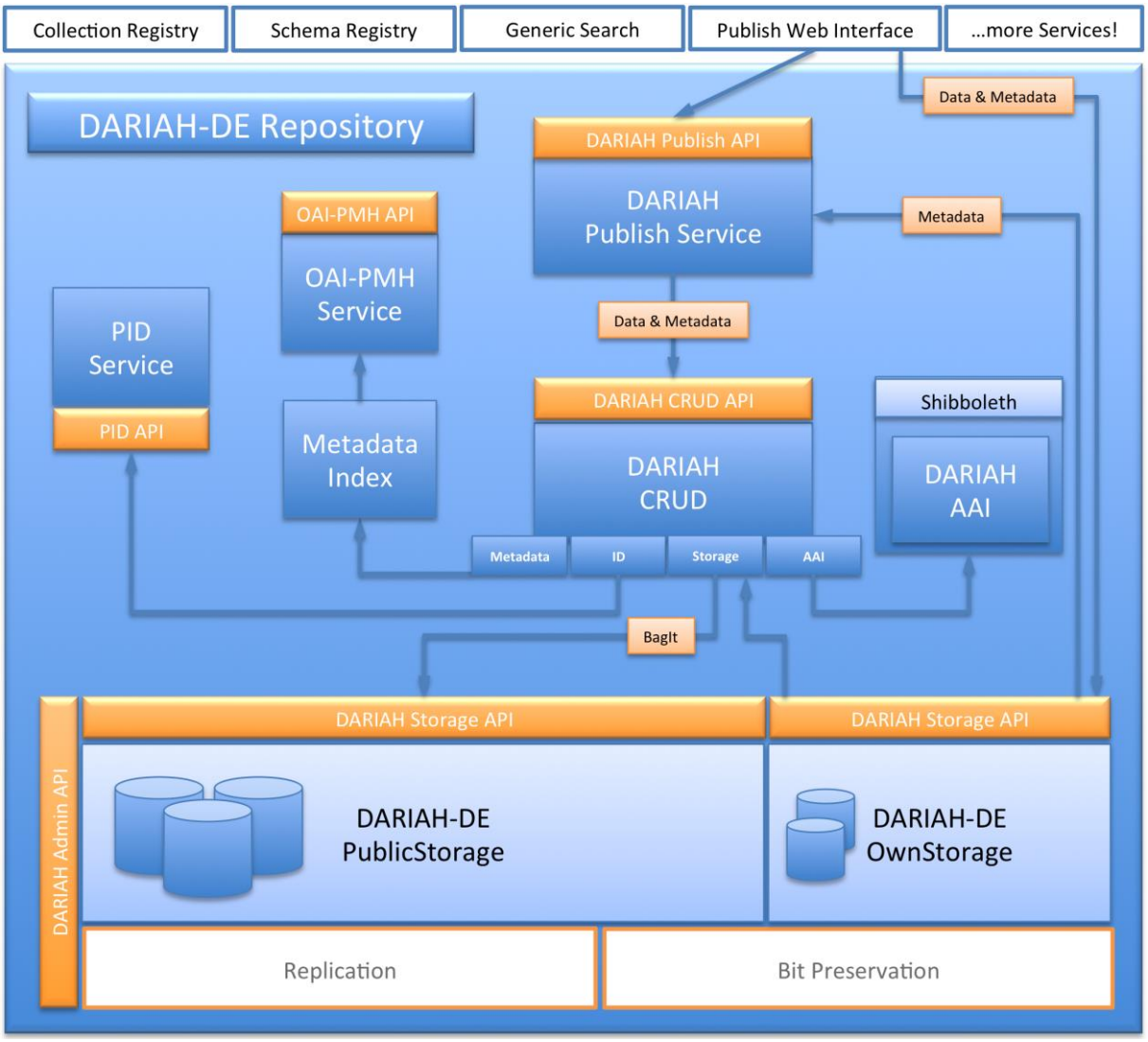

Abbildung 3: Architektur des DARIAH-DE Repositorium Prototyps

2.1. Publish Web-Interface (Publish GUI)

Die Nutzerin oder der Nutzer erzeugt über das DARIAH Publish Web-Interface, implementiert als Liferay-Portlet, eine Kollektion, wählt einzuspielende Daten aus und versieht jedes einzelne Objekt, auch oder gerade die Kollektion selbst, mit DC-Metadaten[13]. Das Web-Interface (oder ein anderer automatisierter Client) liefert die Objekte samt Metadaten per API an den DARIAH-publish Service. Die Dateien werden von der Publish GUI in den OwnStorage – eine Implementierung der DARIAH Storage API[14] – von DARIAH gespeichert, auf den zunächst nur der je-

[13] Zunächst werden hier DC-Simple Metadaten verwendet, die aus 15 Feldern bestehen, später können auch weitere Metadatenschemata unterstützt werden, sofern das Metadaten-Interface für diese implementiert werden. Vgl. Dublin Core Metadata Element Set, Version 1.1. http://www.dublincore.org/documents/dces/

[14] Vgl. DARIAH Storage API – A Basic Storage Service API on Bit Preservation Level. http://hdl.handle.net/11858/00-1734-0000-0009-FEA1-D

weilige Nutzer Zugriff hat. Eine Datei mit allen nötigen Daten und Metadaten wird von der Publish GUI an den Publish Service weiter gegeben (bzw. eine Referenz in den OwnStorage). Die einzuspielenden Dateien können lokal vom Rechner der Nutzerin/des Nutzers stammen, oder auch von einem beliebigen Cloud-Speicher und diese können per Drag and Drop ausgewählt werden.

Der Rückgabewert des Publish Service gibt Aufschluss über den Status des Publikationsvorgangs. Alle möglichen Status sind:

- DRAFT – neu angelegt / in Bearbeitung innerhalb der Publish GUI,
- RUNNING – Publikation ist gerade in Arbeit,
- ERROR – Fehler beim Publikationsprozess,
- PUBLISHED – im DARIAH-DE Repositorium publiziert, und
- REGISTERED – in der Collection Registry registriert und von der generischen Suche indiziert.

Die Publish GUI liefert nach erfolgreicher Publikation einen Link auf die Collection Registry, sowie auf den PID der Kollektion.

2.2. DARIAH-publish Service

DARIAH-publish ist ein Workflow-Service, der verschiedene Schritte im Rahmen der Publikation ausführt. Es werden u. a. die Metadaten validiert, Referenzen auf Objekte innerhalb der einzuspielenden Kollektion von Dateipfaden auf Identifier umgeschrieben und technische Metadaten generiert. Schließlich werden, nach dem Erzeugen der Kollektions-Datei, alle referenzierten Daten samt Metadaten aus dem OwnStorage an den DARIAH-crud weitergegeben.

Wird der Aufruf des Publish Services erfolgreich beendet, ist die Kollektion des Nutzers erfolgreich publiziert worden. Dies bedeutet zunächst, dass

- alle Dateien in den PublicStorage geschrieben wurden, wo sie öffentlich zugänglich sind,
- alle Dateien einen PID[15] erhalten haben,
- die Kollektion und ihre Inhalte über den DARIAH OAI-PMH-Service abfragbar sind, und
- für die anlegende Nutzerin eine Kollektion als Entwurf in der Collection Registry angelegt wurde. Diese muss vom Nutzer noch ergänzt und dort veröffentlicht werden, damit dann die Kollektion von der Generischen Suche per DARIAH OAI-PMH indiziert werden kann. Erst dann sind die Daten auch über die Generische Suche recherchierbar.

[15] Als PIDs werden hier die Handles des EPIC-Konsortiums genutzt (EPIC API v2), vgl. http://www.pidconsortium.eu/ und http://epic.gwdg.de/wiki/index.php/EPIC:API

2.3. DARIAH-crud Service

Der DARIAH-crud Service ist der Speicher-Service des DARIAH-DE Repositoriums, und stellt, genau wie der TG-crud Service, grundlegende Speicher-Operationen zur Verfügung (CREATE, RETRIEVE, UPDATE, und DELETE). Es sind zwei Instanzen des DH-crud Services in Betrieb. Die eine ist nur intern zu erreichen (z. B. vom DARIAH-publish Service) und ist vornehmlich für die Erzeugung und Verwaltung von Daten zuständig (CREATE und evtl. DELETE für administrative Zwecke). Hier werden die Metadaten und Daten aller Objekte

- im DARIAH PublicStorage gespeichert,
- die Metadaten in die Indexdatenbank ElasticSearch eingetragen für einen späteren Abruf per OAI-PMH Service, und
- ein PID erzeugt, der jedes Objekt eindeutig und dauerhaft identifiziert und referenziert.

Die zweite Instanz, die nur lesenden Zugriff auf die Daten erlaubt, ist von extern zu erreichen und gibt Daten- sowie Metadaten der gespeicherten Objekte heraus (READ und READMETADATA).

2.4. Collection Registry

Die Publish-GUI sendet bei erfolgreichem Aufruf des Publish Services den Metadatensatz der Kollektion an die Collection Registry. Dort wird ein Entwurf für den jeweiligen Nutzer angelegt, den der Nutzer komplettieren und als Kollektion veröffentlichen kann. Dieser Schritt ist zum Einen nötig, um dem Nutzer die vollständige Kontrolle über die Registrierung der Kollektion zu geben (und damit die Entscheidung, diese über die Generische Suche verfügbar zu machen), zum Anderen sind verschiedene Angaben zur Kollektion nötig, die nicht schon in der Publish GUI abgefragt, bzw. nicht automatisiert an die Collection Registry weitergegeben werden können.

2.5. Generische Suche

Sobald die Kollektion in der Collection Registry von der Nutzerin/dem Nutzer fertig beschrieben und veröffentlicht wurde (dort wird u. A. die URL zur OAI-Schnittstelle festgelegt), kann die Generische Suche die Daten indexieren und über die Webseite recherchierbar machen. Der OAI-PMH Data Provider kann öffentlich nach neuen Datensätzen des DARIAH-Repositoriums (nach dem OAI-PMH Protokoll) angefragt werden. Dieser nutzt für seine Antworten den ElasticSearch-Index, der von DARIAH-crud Service gefüllt wird. So kann die Generische Suche alle Daten des Repositoriums indexieren und allen Nutzerinnen und Nutzern zur Verfügung stellen. Es werden nur die Daten indiziert, die in öffentlichen Kollektionen der Collection Registry publiziert sind.

3. Publizieren mit dem Prototyp

3.1. Der Publikationsvorgang

3.1.1. Anlegen, editieren und publizieren einer Kollektion

Einloggen

Die Publish GUI des Prototyps des DARIAH-DE Repositoriums ist über die URL

https://dariah.testportal.dariah.eu/publish-gui

erreichbar. Für die Nutzung der Publish GUI ist ein DARIAH-Account nötig, der über das Self-Service-Portal von DARIAH beantragt werden kann:

http://auth.dariah.eu/cgi-bin/selfservice/ldapportal.pl

Sobald der Nutzer sich angemeldet hat, sieht die Oberfläche aus wie in Abbildung 4 (abgesehen vielleicht von den vielen Testkollektionen des Autors):

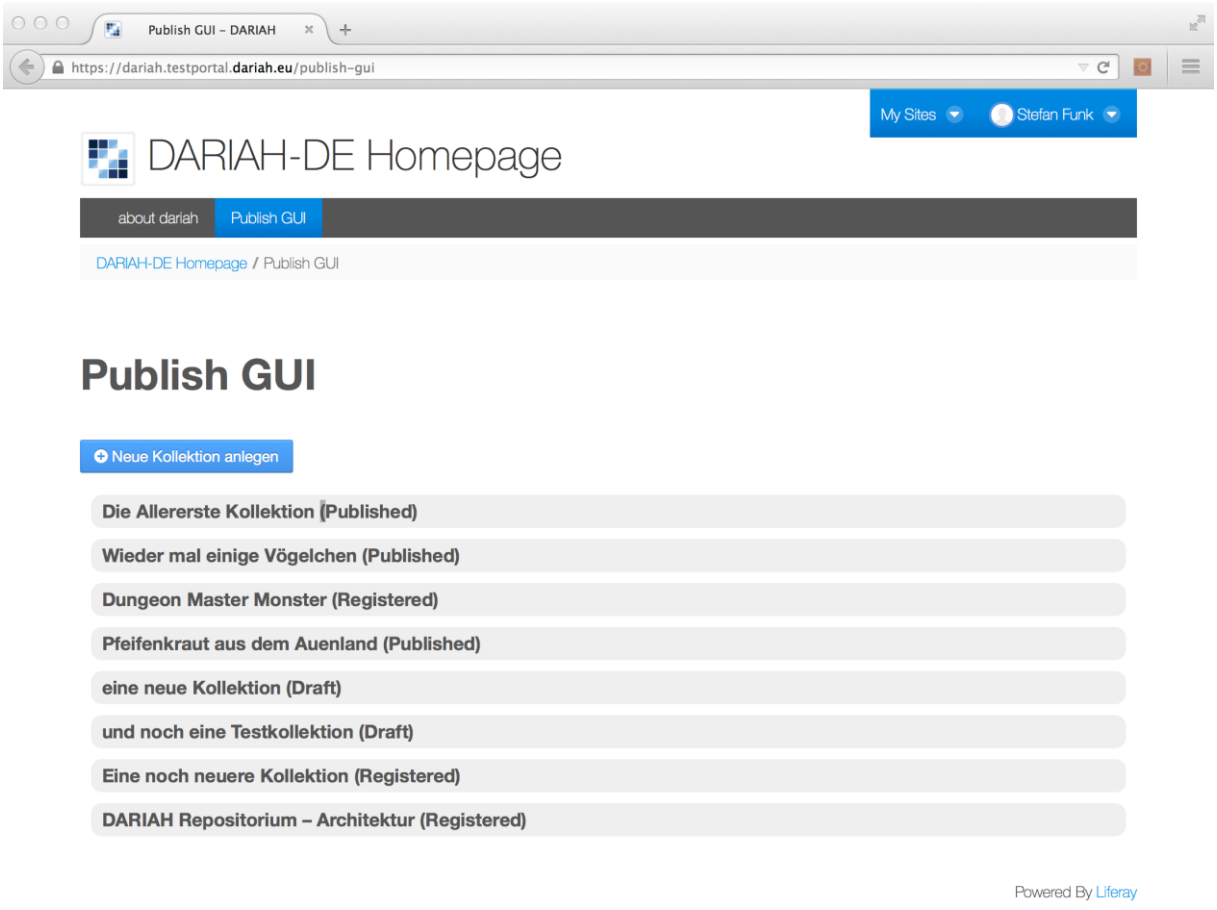

Abbildung 4: Die Publish GUI nach dem Einloggen

Anlegen einer neuen Kollektion

Der erste Schritt hin zu einer Publikation ist das Anlegen einer neuen Kollektion durch Klick auf den Button „Neue Kollektion anlegen“, das Bearbeiten eines Entwurfs ist hier später ebenfalls möglich. Es erscheint die Ansicht zum Eingeben der Kollektions-Metadaten sowie zum Hinzufügen von Dateien (Abbildung 5).

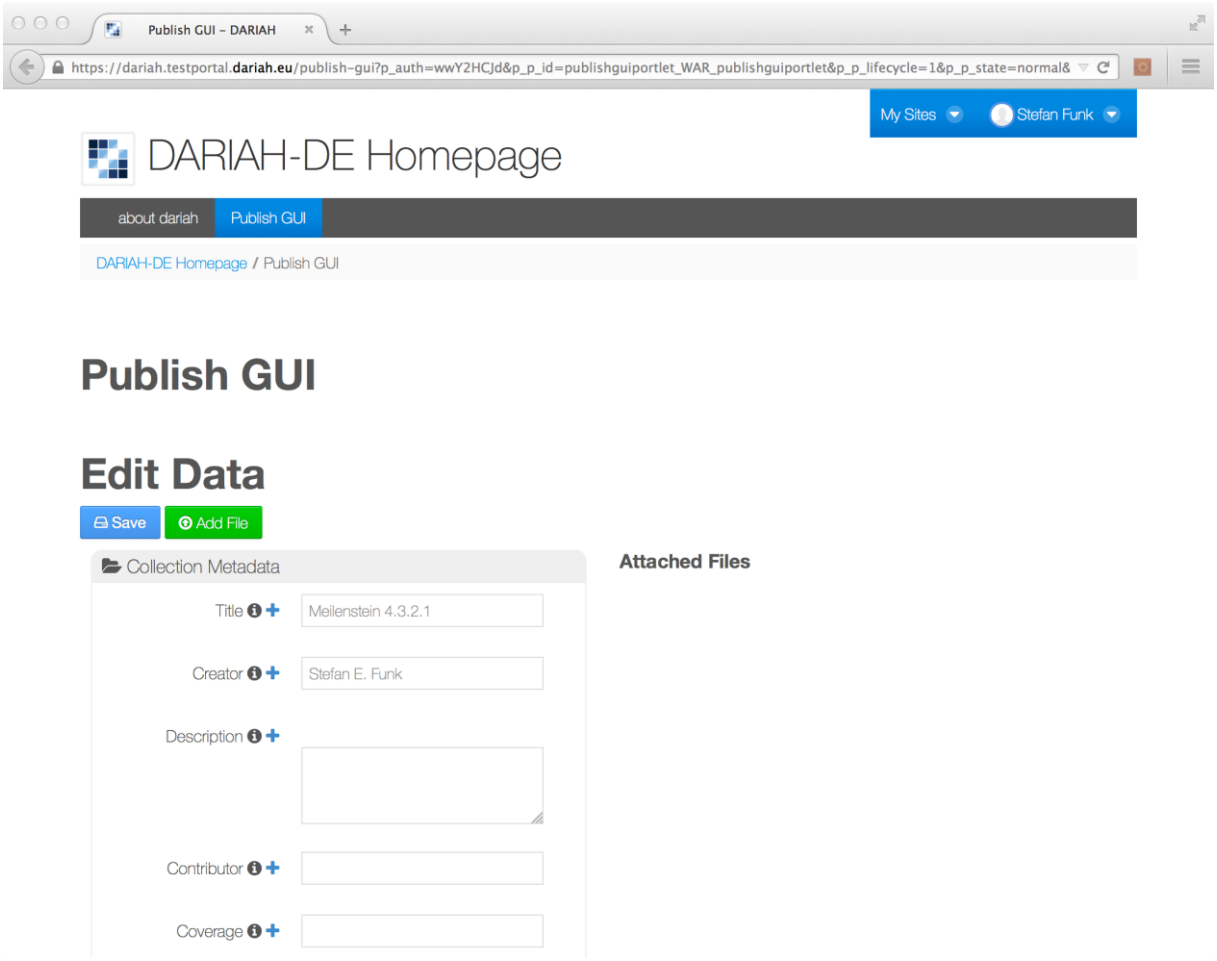

Abbildung 5: Eingeben der Kollektions-Metadaten

Nach Eingabe eines Titels (title) und Autors (creator) kann die Kollektion gespeichert werden und erscheint später in der Übersicht unter dem eingegebenen Titel (Abbildung 6). Weitere Metadaten sind beim Prototyp für eine Publikation nicht erforderlich. Es können jedoch alle DC-Metadaten durch die Nutzerin/den Nutzer (auch mehrfach) vergeben werden.[16]

[16] In den weiteren Entwicklungsschritten der Publish GUI werden noch folgenden Funktionalitäten eingebaut: Löschen von Kollektionen, Löschen von DC-Metadaten-Feldern, etc. Weitere Funktionalitäten werden auf Anforderungsbasis zukünftig hinzugefügt.

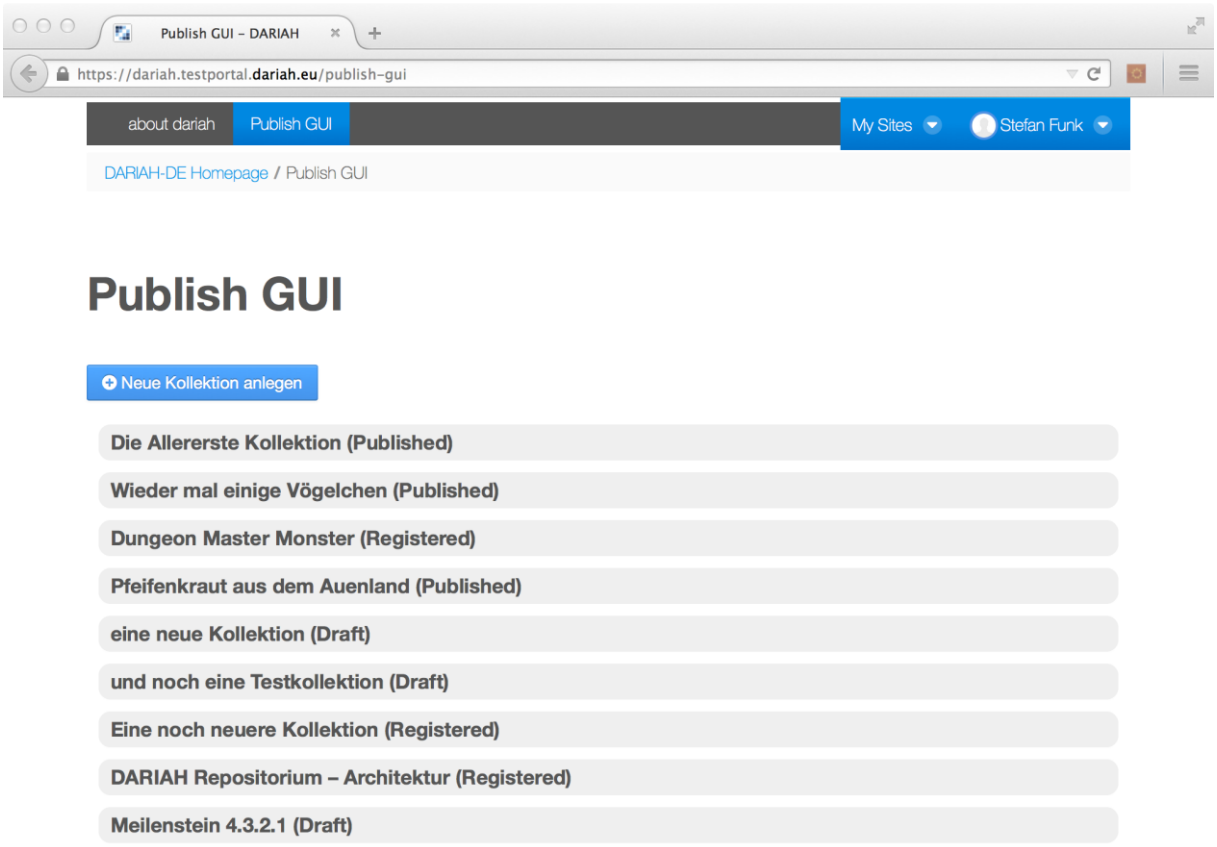

Abbildung 6: Die neu angelegte Kollektion in der Übersicht

Anhängen von Dateien

Nachdem die Kollektion beschrieben wurde, können per Klick auf „Datei hinzufügen“ Dateien an diese angehängt werden. Dies geschieht über das Browser-eigene Upload-Fenster. Die Dateien werden zunächst in den DARIAH OwnStorage hochgeladen, auf den zunächst nur der im Portal angemeldete Nutzer Zugriff hat. Abbildung 7 zeigt die GUI nach dem Hochladen von drei Bildern.

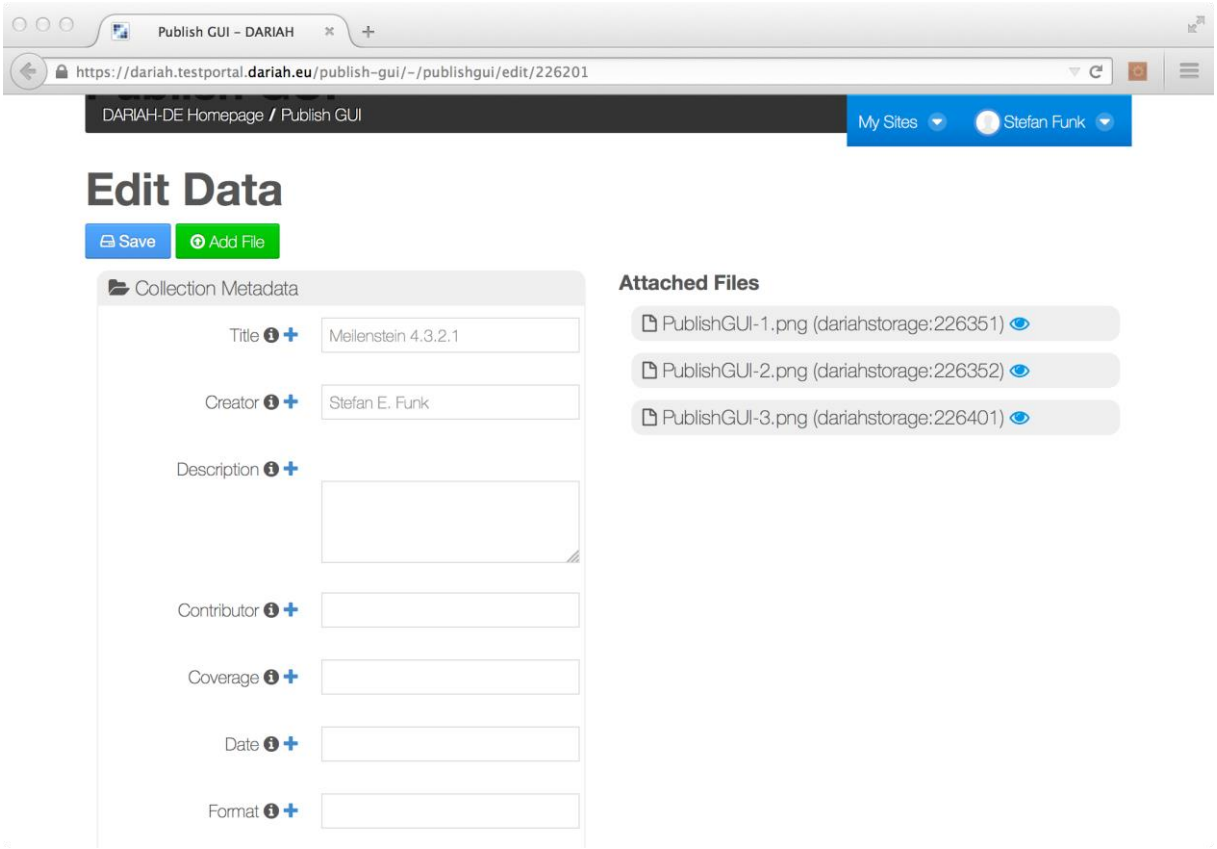

Abbildung 7: Zur Kollektion zugehörige Dateien nach dem Hinzufügen

Nun können die Titelleisten der einzelnen Bilder angeklickt werden. Es erscheint ein Metadatenformular zur Eingabe von Metadaten für die jeweilige Datei. Der Dateiname wird zunächst als Titel angenommen, das Format der Datei wird automatisch ermittelt und eingetragen. Das Ändern des Titels sowie die Vergabe von weiteren Metadaten ist nun möglich, wie in Abbildung 8 abgebildet.

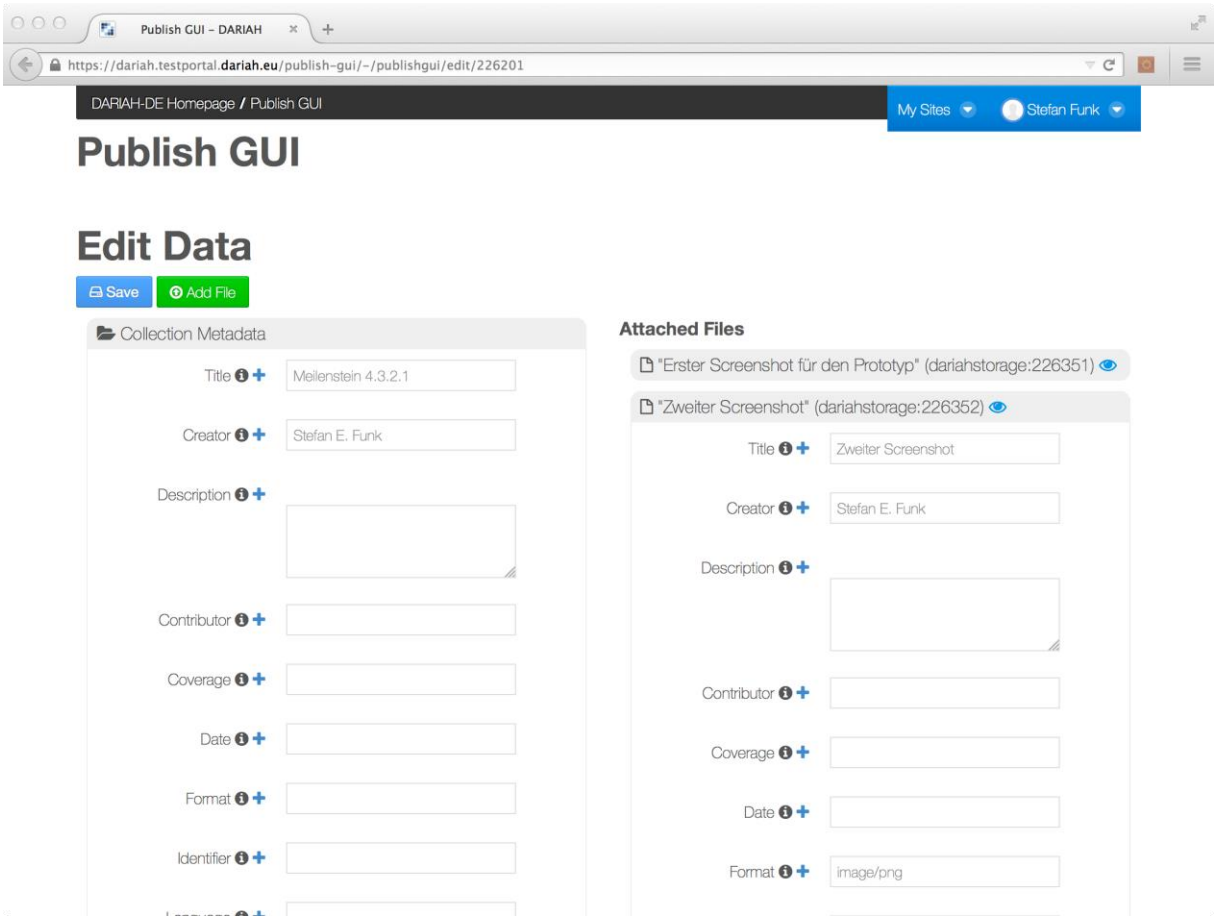

Abbildung 8: Eingeben von Metadaten für einzelne Dateien

Speichern der Daten und Rückkehr zur Übersicht

Sind alle zur Kollektion nötigen Dateien hinzugefügt und alle benötigten Metadaten vergeben (bzw. soll das Editieren der Kollektion unterbrochen werden), kann die Kollektion per Klick auf den „Speichern“-Button gesichert werden. Es werden nun alle Metadaten samt der Referenzen auf die in OwnStorage gespeicherten Dateien ebenfalls im OwnStorage abgelegt. So sind die Daten und Metadaten der Kollektion gesichert und können zu einem späteren Zeitpunkt weiter bearbeitet werden. Nach Klick auf „Publish GUI“[17] gelangt man zurück zur Übersicht seiner Kollektionen (Abbildung 9).

[17] bzw. den noch zu implementierenden „Zurück zur Übersicht“-Button

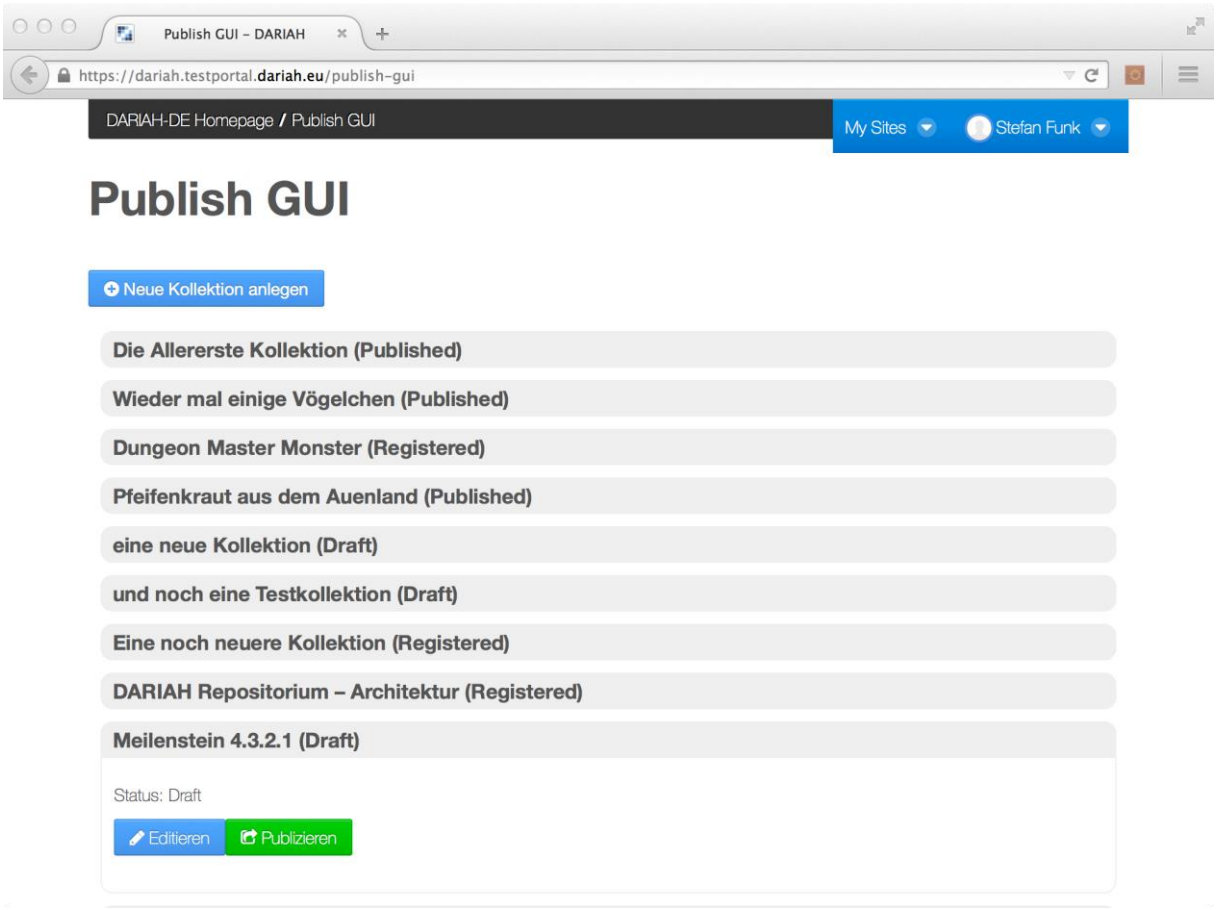

Abbildung 9: Die angelegte Kollektion in der Übersicht

Publikation der Kollektion

Soll die Kollektion weiter bearbeitet werden, können per Klick auf „Editieren“ der Kollektion weitere Metadaten sowie weitere Dateien hinzugefügt werden, sowie auch die vorhandenen Metadaten editiert werden. Ist die Kollektion fertig bearbeitet, kann sie nun endgültig durch Klick auf „Publizieren“ publiziert werden. Dies bedeutet, dass die Daten und Metadaten in den öffentlichen Bereich des DARIAH-Storage (PublicStorage) geschrieben werden und nach dem Publizieren nicht mehr geändert und auch nicht mehr gelöscht werden können, auch nicht mehr von dem Nutzer, der die Kollektion publiziert hat. An dieser Stelle ist eine gute Dokumentation für die Nutzer nötig, um zu kommunizieren, welche Konsequenzen eine solche Publikation in das DARIAH-DE Repositorium hat. Es wird sicherlich auch notwendig sein, den Nutzer aufzufordern, eine Lizenz für seine Daten zu vergeben, damit die Möglichkeiten einer Nachnutzung der Daten dieser Kollektion dokumentiert sind. Dies wird momentan im Rahmen von DARIAH-DE diskutiert und wird im produktiven DARIAH-DE Repositorium implementiert werden.

Nach Klick auf den Publizieren-Button wird der Publikationsprozess angestoßen und der DARIAH-publish Service beginnt, wie in Kapitel 5.1 beschrieben, die Veröffentlichung der Daten wie dort beschrieben abzuarbeiten. Beim Aktualisieren der Übersicht in der Publish GUI bekommt man nun den Status des Publikations-

vorgangs angezeigt, alle möglichen Status sind in Kapitel 2.1 beschrieben. Nach einer erfolgreichen Publikation – während dessen der Status samt einer Prozentangabe des Fortschritts auf RUNNING gesetzt ist –, wird der Status PUBLISHED an die Publish GUI zurück geliefert (siehe Abbildung 10 und Abbildung 11).

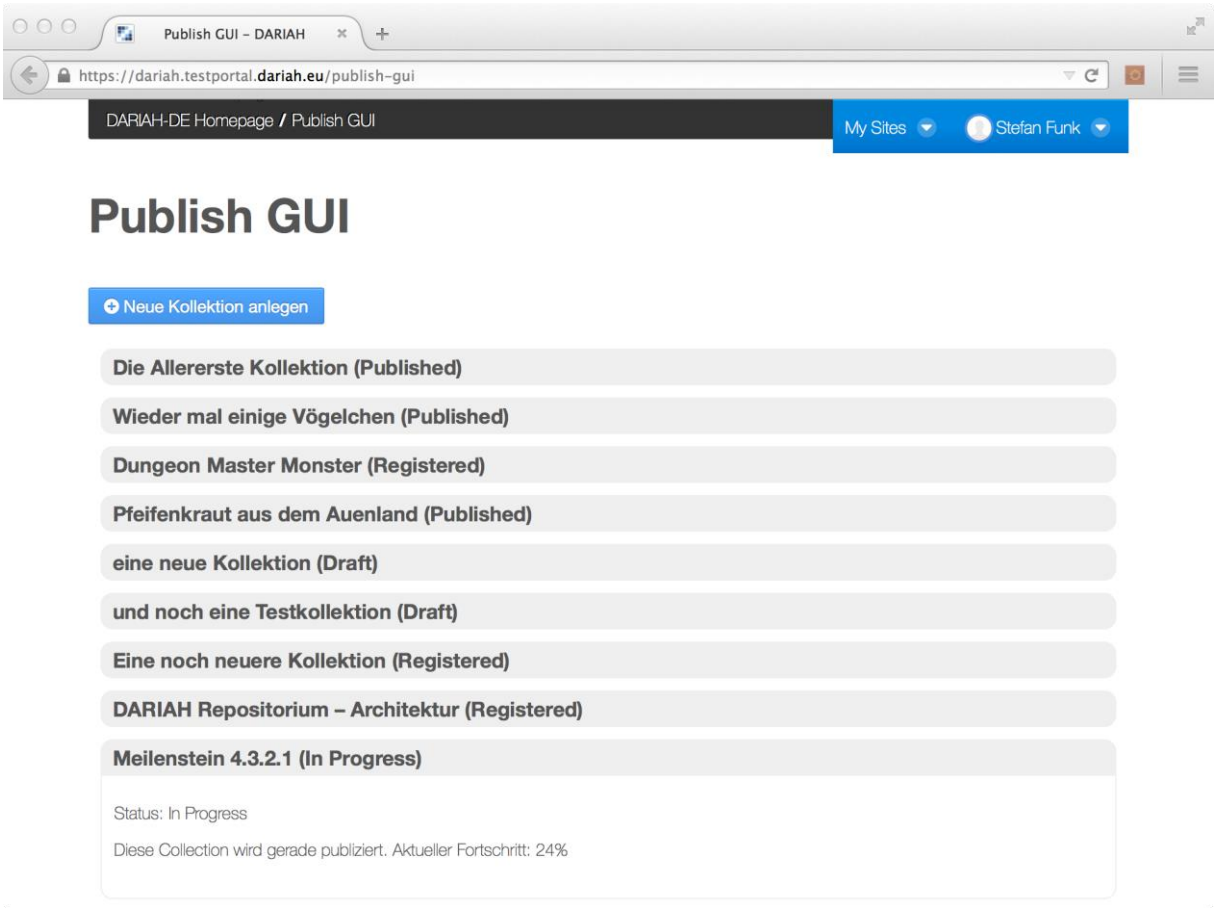

Abbildung 10: Publikation in Progress

Es kann nun mit der Registrierung der Kollektion in der Collection Registry fortgefahren werden. Im Fehlerfall erhält man den Status ERROR, der Fehler kann angezeigt und evtl. durch erneutes Editieren der Metadaten durch den Nutzer behoben werden. Dies kann beispielsweise geschehen, wenn alle verpflichtenden Metadaten eingegeben wurden.

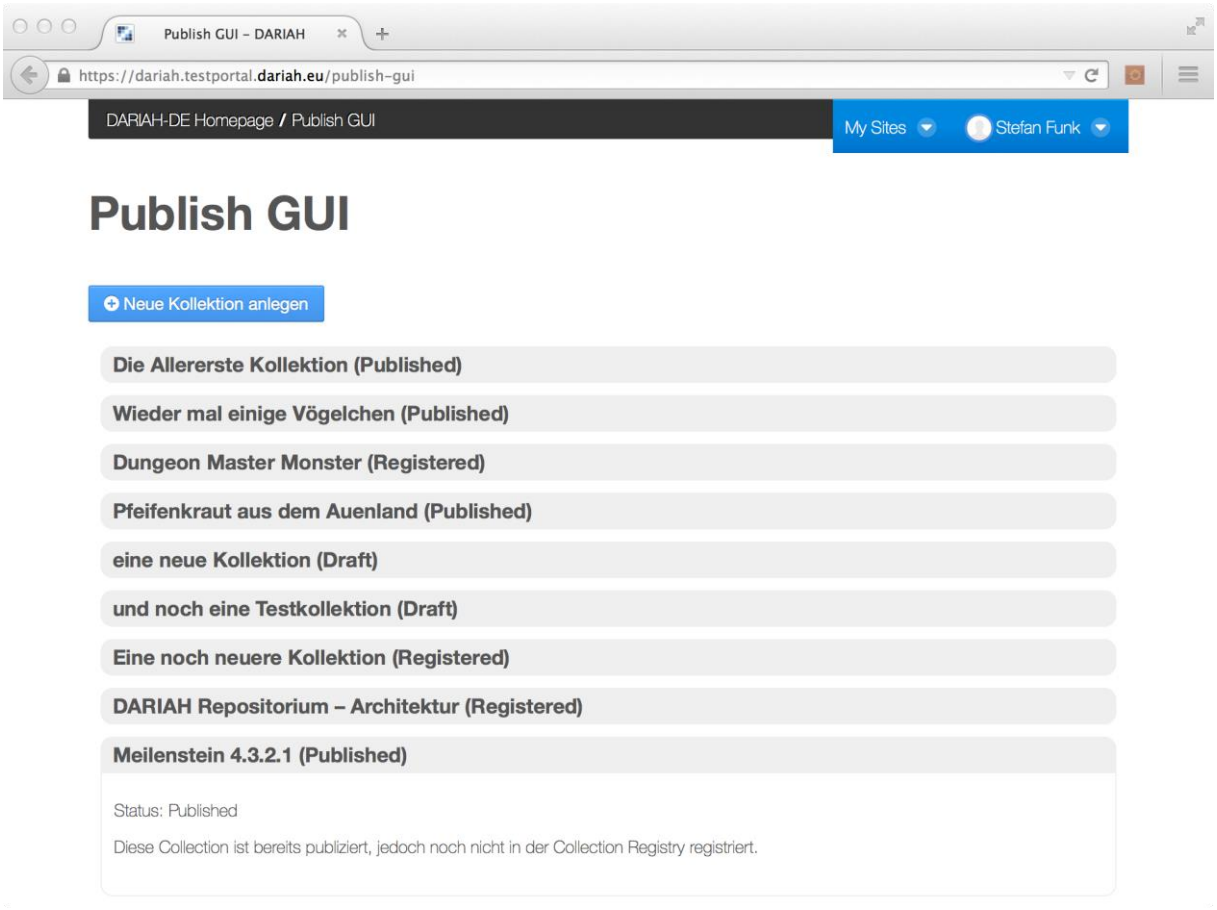

Abbildung 11: Erfolgreich publizierte Kollektion

Persistent Identifier

Ist eine Kollektion erfolgreich per Publish GUI publiziert worden, hat die Kollektion selbst (als übergeordnetes Objekt) sowie jede dazugehörige Datei einen Persistent Identifier (PID) bekommen, mit der die Kollektion und ihre Dateien dauerhaft und eindeutig referenzierbar sind. Unsere Testkollektion „Meilenstein 4.3.2.1“ hat z. B. den PID 11022/0000-0000-4F2E-6. Auflösbar ist diese PID über jeden Handle-Resolver, wie beispielsweise

http://hdl.hande.net oder auch http://dx.doi.org.

Die Metadaten für diesen PID sind einsehbar per

http://hdl.handle.net/11022/0000-0000-4F2E-6?noredirect

oder auch via

http://dx.doi.org/11022/0000-0000-4F2E-6?noredirect,

da es sich bei den von DARIAH-DE genutzten Handles technisch gesehen um DOIs handelt (bzw. bei den DOIs technisch gesehen um Handles). Lässt man das „?noredirect“ am Ende der URL weg, wird man direkt weitergeleitet zum jeweiligen Objekt, das hinter der PID steht, in unserem Fall zu der TAR-Datei, die unsere Kollektion samt deren Metadaten enthält (siehe Abbildung 12).

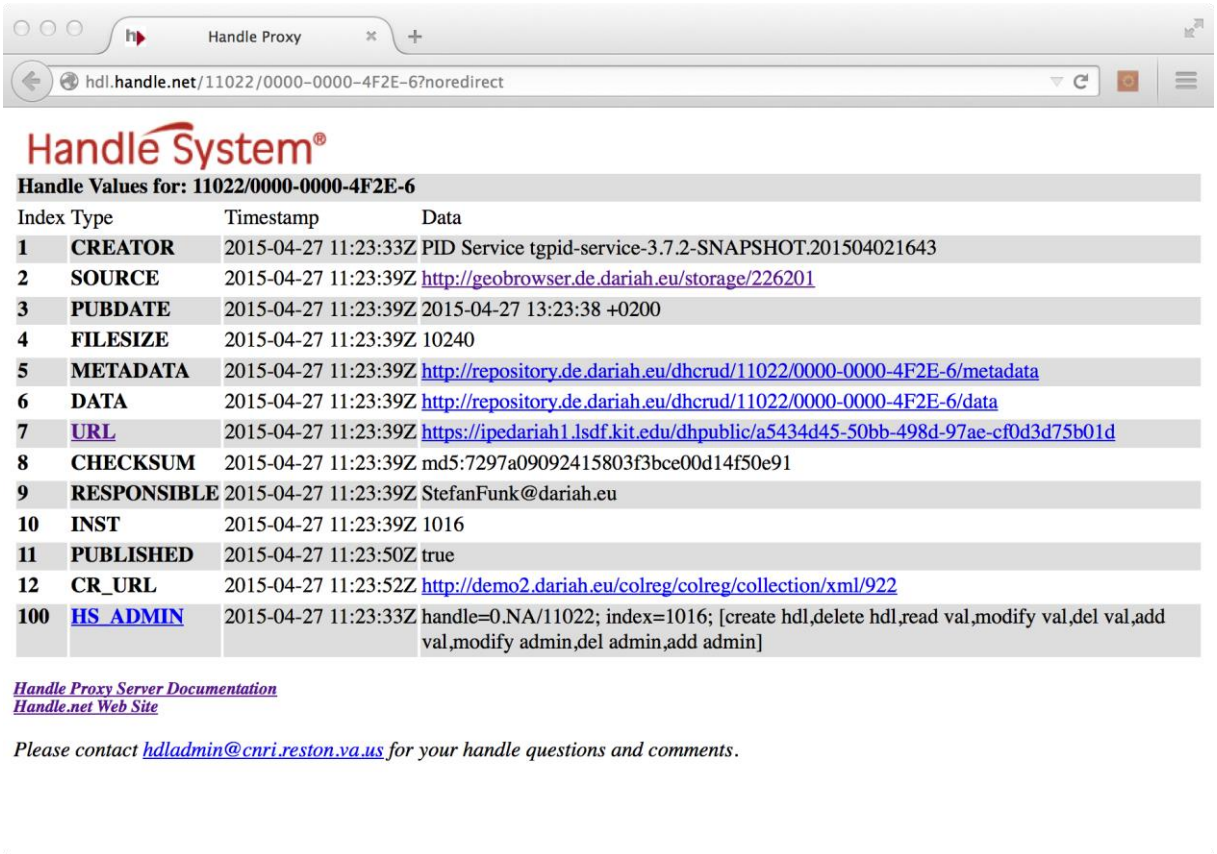

Abbildung 12: Die Metadaten eines Handle PIDs

3.1.2. Registrieren der Kollektion in der Collection Registry

Die DAHRIAH-DE Collection Registry (CR) für den Repositoriums-Prototyp ist erreichbar unter

https://demo2.dariah.eu/colreg

und verzeichnet momentan nur Testkollektionen, die für den Prototyp eingetragen wurden. Im letzten Schritt des Publikationsprozesses via DARIAH-publish werden einige Metadaten an die Collection Registry gesendet (u. A. der Titel der Kollektion und eine Kennung zur Identifizierung des jeweiligen Nutzers), so dass zunächst ein Entwurf einer Kollektionsbeschreibung in der CR angelegt wird, die nur für den Nutzer selbst sichtbar ist.

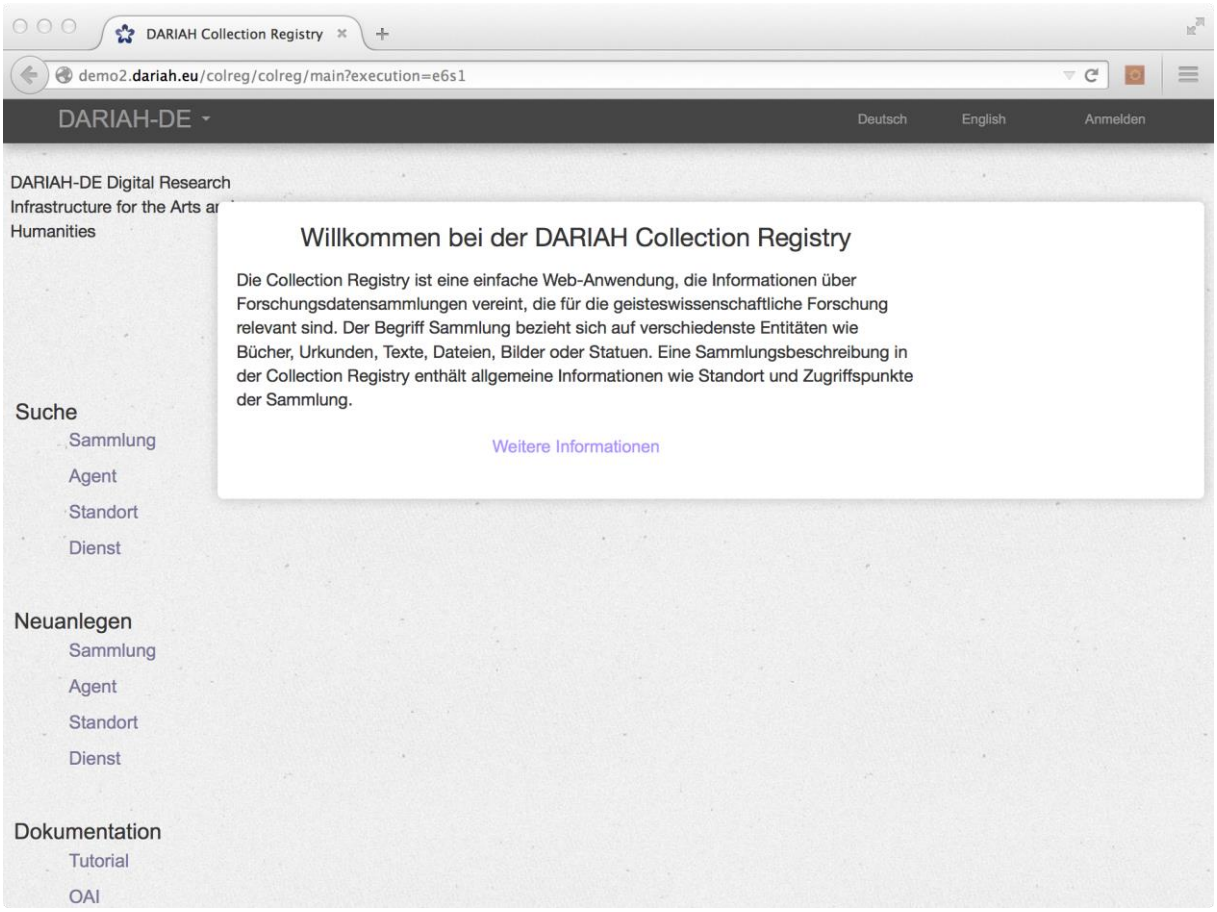

Abbildung 13: Die Startseite der Collection Registry

Der Nutzer wird nun in der Collection Registry mit denselben Credentials angemeldet, mit denen er auch in der Publish GUI angemeldet ist (Single Sign-On via Shibboleth), so dass kein weiterer Login notwendig ist, dies wird alles transparent im Browser gehandhabt.

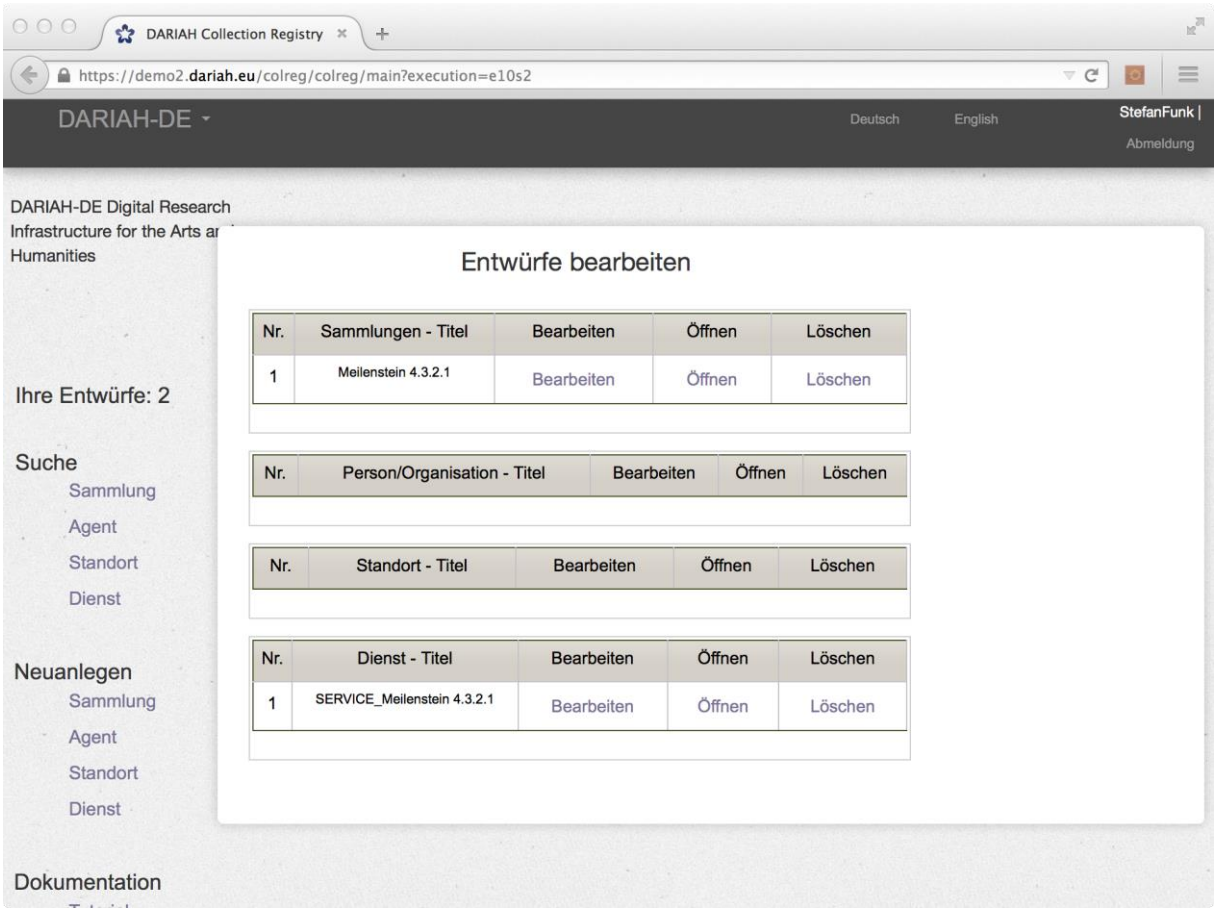

Abbildung 14: Eigene Entwürfe in der Collection Registry

Mit einem Klick auf „Ihre Entwürfe“ werden dann schließlich zwei Entwürfe angezeigt: Ein Entwurf für einen Dienstbeschreibung und ein Entwurf für eine Sammlungsbeschreibung. Die Beschreibung einer Kollektion in der Collection Registry beruht auf einem komplexen Schema, zu dem u. A. ein Dienst gehört, unter dem die Metadaten der Kollektion und ihrer Inhaltsdateien per OAI-PMH abgerufen werden können. Auf diese OAI-PMH-Metadaten stützt sich dann die Indexierung der Generischen Suche. Die URL für diese OAI-PMH-Schnittstelle des DARIAH-DE Repositoriums und eine ID der Kollektion wird ebenfalls schon vom DH-publish Service an die CR übergeben.

Einrichten des Dienstes für die publizierte Kollektion

Durch Klick auf „Bearbeiten“ beim Dienste-Entwurf (siehe Abbildung 14, Dienst „SERVICE_Meilenstein 4.3.2.1“) können die Einzelheiten des Dienstes eingesehen und nun auch editiert werden. Da hier jedoch bereits alle erforderlichen Daten vom DARIAH-Publish Service übermittelt wurden, kann dieser Dienst einfach per Klick auf „Beenden“ und „Speichern und Veröffentlichen“ publiziert werden. Dieser Dienst ist bereits automatisch der Kollektionsbeschreibung zugeordnet.

Einrichten der Kollektionsbeschreibung

Um den Entwurf der Kollektionsbeschreibung zu komplettieren und zu veröffentlichen, ist durch Klick auf „Bearbeiten“ des übrig gebliebenen Entwurfs (Abbildung

14, Sammlung „Meilenstein 4.3.2.1“) das Bearbeiten der Kollektionsbeschreibung möglich. Als einzige nicht übermittelte benötigte Daten sind hier der Eigentümer sowie der Standort der Sammlung zu nennen. Beide können nicht automatisch übermittelt werden, da es sich nicht um einfache Zeichenketten, sondern eigenständige Objekte innerhalb der Collection Registry handelt, die auch weiterhin für die Beschreibung neuer Kollektionen genutzt werden können. Der Titel sowie der zugehörige OAI-PMH-Dienst sind bereits in die Beschreibung der Kollektion übernommen worden. Eigentlich müssen nun die erforderlichen beiden Metadaten (Eigentümer und Standort) durch diverse Klicke auf „Weiter“ eingetragen, bzw. erstellt und ausgewählt werden. Durch Klick auf „Speichern und Veröffentlichen“ schließlich wird auch die Kollektionsbeschreibung veröffentlicht werden. Nun kann die Kollektion von der Generischen Suche indexiert und nachgewiesen werden.[18]

Dieser Status des Publikationsvorgangs wird nun auch in der Publish GUI angezeigt (siehe Abbildung 15).

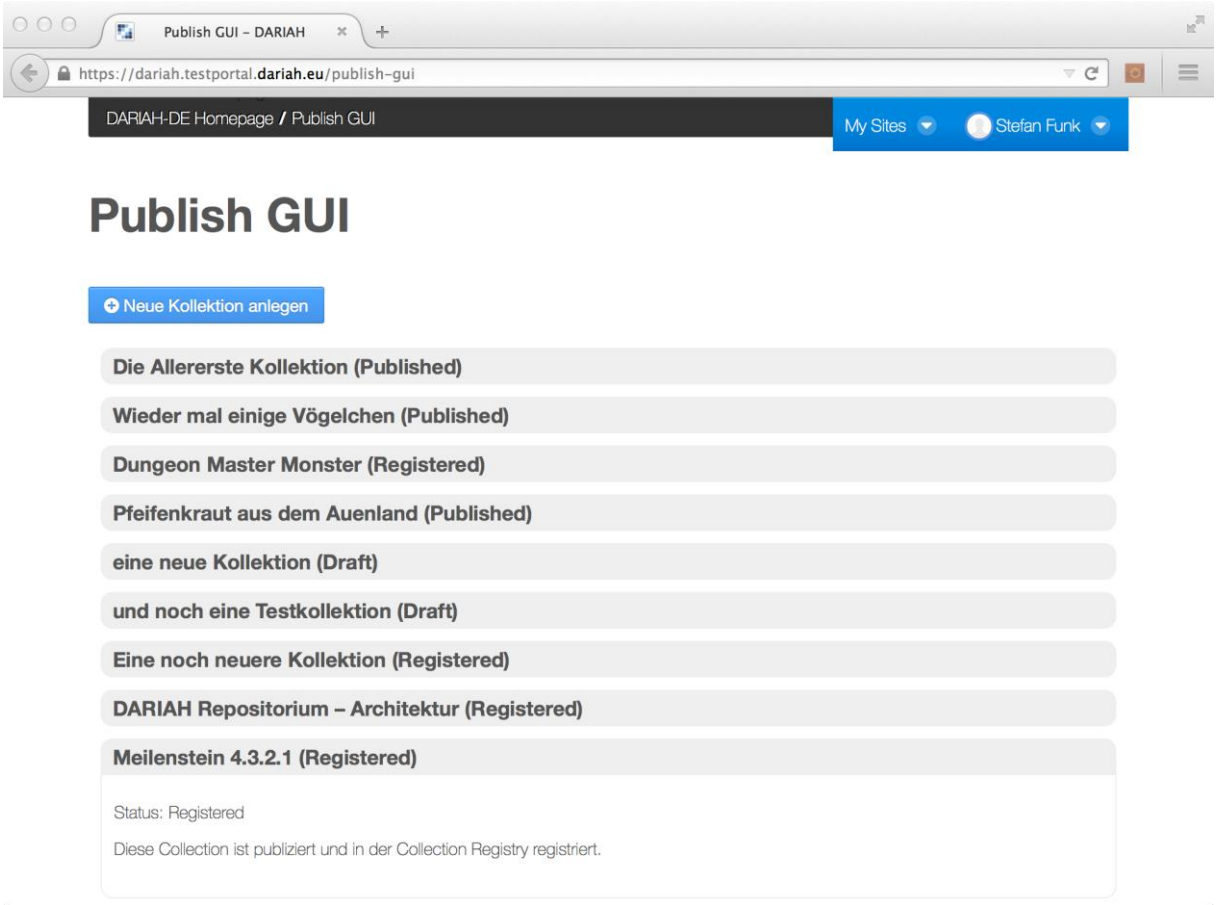

Abbildung 15: Status „REGISTRIERT“

[18] Die Collection Registry wird zur Zeit überarbeitet und technisch wie auch inhaltlich an neue Anforderungen angepasst. So soll z. B. die Generische Suche von der CR gemeldet bekommen, wenn neue Kollektionsbeschreibungen publiziert wurden, so dass die GS nicht alle 24 Stunden alle Kollektionen neu indexieren muss, sondern nur die neu hinzugekommenen.

3.1.3. Nachweis und Suche mit der Generischen Suche (GS)

Sobald eine Kollektionsbeschreibung mit der Collection Registry publiziert wurde, werden die Daten, die per OAI-PMH seit dem Abschließen des Publikationsvorgangs aus der Publish GUI verfügbar sind, von der Generischen Suche indexiert. Dies passiert zunächst automatisch alle 24 Stunden, und ab dann sind alle Objekte sowie die Kollektion selbst über die Generische Suche zugreifbar. Die Generische Suche des DARIAH-DE-Repositoriums ist unter

http://dev3.dariah.eu/search/

zu erreichen. Dort sind die Inhalte aller prototypischen Kollektionen indexiert und können durchsucht werden.

Abbildung 16: Beispielansicht der Generischen Suche

3.2. Der Prototyp in URLs

Der Prototyp des DARIAH-Repositoriums enthält die nachfolgenden Komponenten und Funktionalitäten:

3.2.1. Dienste

- DARIAH Publish GUI – Web-basiert als Portlet im DARIAH-Portal: https://dariah.testportal.dariah.eu/publish-gui/
- DARIAH-publish – Web-Service im Tomcat: http://repository.de.dariah.eu/dhpublish/
- DARIAH-crud – Web-Service im Tomcat (incl. PID-Identifier- und DC-Metadaten-Modul): http://repository.de.dariah.eu/dhcrud/ (read-only für öffentlichen Zugriff auf die Daten)
- ElasticSearch Metadata Index – Standalone Application Server (interner Zugriff)
- DARIAH OAI-PMH-Service – Web-Service im Tomcat: http://repository.de.dariah.eu/oaipmh/oai?verb=Identify

3.2.2. Server

- zunächst der DARIAH Repository-Server: http://repository.de.dariah.eu
- Collection Registry: http://demo2.dariah.eu/colreg/
- Generische Suche: http://dev3.dariah.eu/search/
- Handle Services: http://hdl.handle.net (für öffentliche Abfragen) und http://pid.gwdg.de (für Generierung und Schlüssel/Wert-Paar-Verwaltung)

3.2.3. Entwicklung und Monitoring

- Monitoring: https://dariah.fz-juelich.de/cgi-bin/icinga/status.cgi?servicegroup=dh-rep-proto&style=detail
- Issue-Tracking: https://projects.gwdg.de/projects/dariah-de-repository/issues

3.3. URLs einer Beispiel-Kollektion

- DH-publish Info/Status: http://repository.de.dariah.eu/dhpublish/225351/info
- PID: 11022/0000-0000-4EAC-8
- CR-Referenz: https://demo2.dariah.eu/colreg/colreg/collectionDetails?id=908
- PID-Metadaten: http://hdl.handle.net/11022/0000-0000-4EAC-8?noredirect

- PID-Auflösung zum Objekt (Bag):
  http://hdl.handle.net/11022/0000-0000-4EAC-8

- Direkter Link zu den DC Metadaten (siehe PID-Metadatum METADATA):
  http://repository.de.dariah.eu/dhcrud/11022/0000-0000-4EAC-8/readMetadata

- Direkter Link zum Objekt (siehe PID-Metadatum DATA):
  http://repository.de.dariah.eu/dhcrud/11022/0000-0000-4EAC-8/read

- Direkter Link zur Bag (siehe PID-Metadatum URL):
  https://ipedariah1.lsdf.kit.edu/dhpublic/45431a58-670c-4f5f-8847-e62fdce69920

4. Ausblick auf das DARIAH-DE Repositorium

Der Prototyp wird in den kommenden Monaten bis Juli 2015 zunächst vom Konsortium ausgiebig getestet, so dass das Konsortium Use-Cases, die Publikation von Forschungsdaten beinhalten und den Publikationsvorgang auf Funktionalität prüfen und kommentieren können. Insbesondere soll auf diese Weise sichergestellt werden, dass die umgesetzten Publikationsprozesse generischen Charakter aufweisen und u. a. hinsichtlich Usability und der graphischen Nutzerführung in unterschiedlichen disziplinären Kontexten genutzt werden können. Darüber hinaus soll in dieser Phase eruiert werden, welche weiteren Funktionalitäten prioritär implementiert werden.

Zum jetzigen Zeitpunkt deuten sich bereits folgende Prioritäten an. Dies ist zum Einen die Möglichkeit, die Daten in Sub-Kollektionen zu unterteilen, und damit bis zu einem gewissen Grad zu strukturieren. Zum Anderen wird eine Versionierung der Kollektionen und ihrer Daten erforderlich sein, so dass auch bereits publizierte Kollektionen, die zunächst fest mit ihren Daten verknüpft sind und somit nicht gelöscht werden können, durch eine Versionierung gewisser Maßen sowohl aktualisiert werden können, als auch deren Ursprungszustand erhalten bleibt, um deren persistente Referenzierbarkeit sicherzustellen.

Weiterhin wird das DARIA-DE Repositorium weitere Anforderungen an die Authentifizierungs- und Autorisierungs-Infrastruktur stellen, so dass z. B. die Daten aus dem OwnStorage nicht im Namen eines Services mit Funktions-Account gelesen werden (wie es momentan im Prototyp realisiert ist), sondern per Delegation von AAI-Tokens im Namen des Nutzers. Die technischen Einzelheiten hierfür müssen noch evaluiert und in einem weiteren Schritt entwickelt werden. Darüber hinaus ist in Kooperation mit externen Projektpartnern eine exemplarische Anbindung an fachwissenschaftliche Dienste geplant, um zu evaluieren, inwiefern der Publikationsprozess von Forschungsdaten in einem Repositorium an den toolgesteuerten Forschungsprozess unmittelbar gekoppelt werden kann.

5. Technische Grundlagen

Die technische Infrastruktur des DARIAH-DE Repositoriums beruht auf den Services, die in TextGrid für das TextGrid Repository entwickelt wurden und dort auch im produktiven Einsatz sind. Um diese für das DARIAH-DE Repositorium nachnutzen zu können, wurden und sind einige Anpassungen notwendig, um den Quellcode für beide Repositorien gleichsam zu nutzen und zukünftig auch gemeinsam pflegen zu können. Die Grafik in Abbildung 17 zeigt das Zusammenspiel der zwei Repositorien und deren Komponenten, die im Folgenden beschrieben werden.

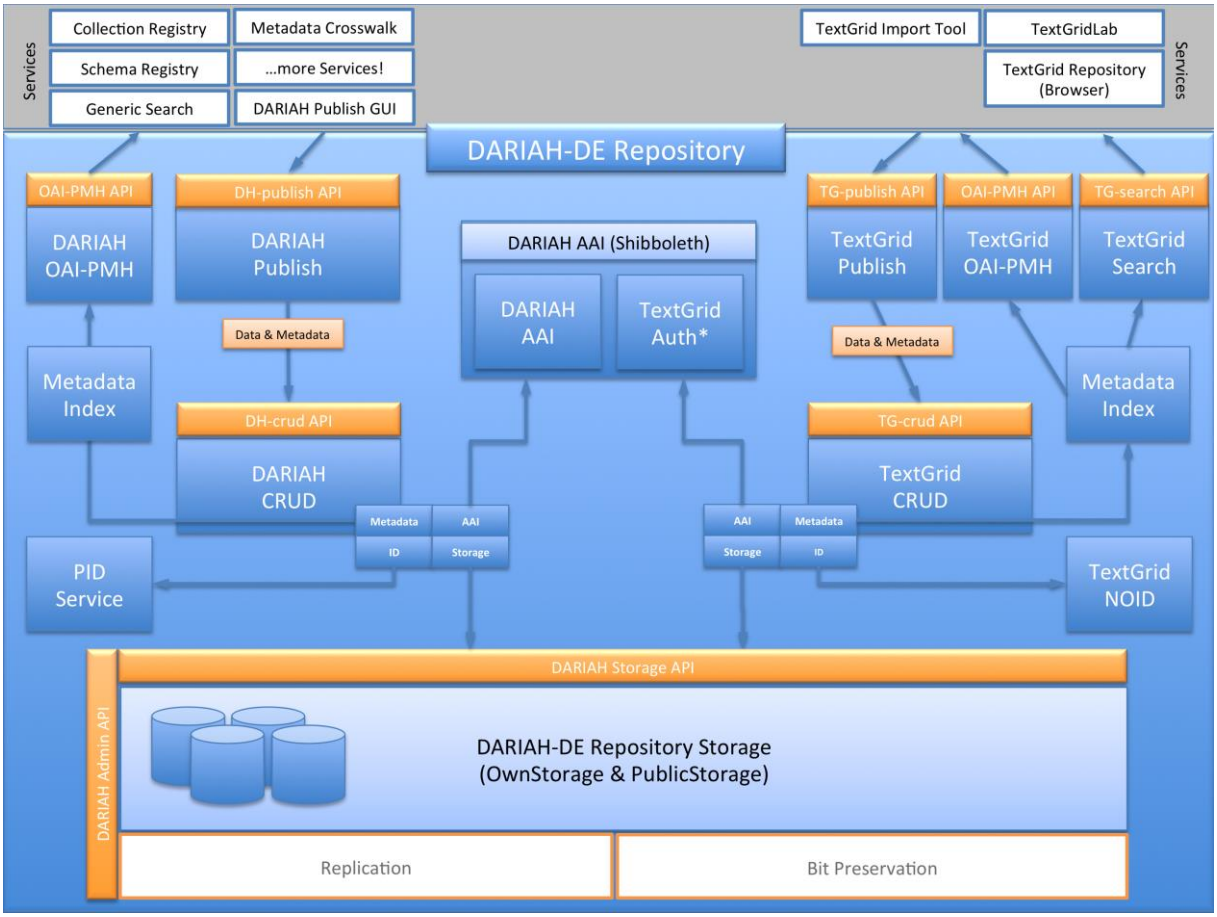

Abbildung 17: Architektur des TextGrid/DARIAH-DE Repositoriums

5.1. DARIAH-publish Service und API

Der DARIAH Publish Service nimmt Daten von diversen DARIAH-Publish Clients an (momentan hauptsächlich von der Publish GUI, später evtl. von weiteren, möglicherweise automatisierten, Clients) und validiert beispielsweise die Metadaten, schreibt Referenzen auf Objekte innerhalb der einzuspielenden Kollektion von Dateipfaden auf Identifier um und generiert technische Metadaten. Weitere Module können problemlos erstellt werden. Der Workflow innerhalb des Publish Ser-

vices wird mittels des Workflow Tools koLibRI[19] implementiert (wie auch der TextGrid Publish Service), und es kommen die folgenden Module zum Einsatz:

- CheckCollection – Die von der Publish GUI übermittelte Referenz auf die RDF-Datei, die alle notwendigen Daten und Metadaten enthält, wird aufgelöst und die Datei aus dem DARIAH OwnStorage heruntergeladen (auf den der Service lesenden Zugriff hat), die Metadaten werden geprüft.
- CreateCollection – Eine Kollektions-Datei und deren Metadaten werden aus der RDF-Datei extrahiert und per DARIAH-crud#CREATE im Public-Storage angelegt.
- MetadataProcessor – Es werden die Metadaten aller Dateien bearbeitet und für den Publikationsprozess benötigte Metadaten ergänzt.
- SubmitFiles – Alle zur Kollektion zugehörigen Dateien werden per DARIAH-crud#CREATE in den DARIAH PublicStorage geladen. Nach diesem Prozess haben alle Dateien einen Persistenten Identifier (PID).
- NotifyCollectionRegistry – Die Metadaten der Kollektion werden an die Collection Registry übergeben, damit dort ein Entwurf dieser Kollektion angelegt werden kann (unter Anderem auch die URL zur OAI-Schnittstelle des DARIAH-Repositoriums samt Spezifizierung der eben angelegten Kollektion als OAI-PMH Set). Diesen kann dann die Nutzerin/der Nutzer ergänzen und als Kollektion in der Collection Registry publizieren. Sobald die Kollektion hier veröffentlicht ist, wird sie von der Generischen Suche indiziert und nachgewiesen.
- UpdatePidMetadata – Einige Metadaten werden übe den PID-Service gespeichert, zum Beispiel auch die Anzeige, dass eine Kollektion nun korrekt und vollständig publiziert ist.

5.2. Storage-Dienste

Für eine Implementierung des DARIAH-crud Services konnte der Quellcode des TG-crud nachgenutzt werden. Dieser war bereits für alle genutzten Datenbanken und Service-Zugriffe modularisiert, so dass nur einige neue Schnittstellen implementiert werden mussten. Die Schnittstellen zum PID-Service, zur DARIAH Storage-API sowie zum OAI-Service konnten nahezu unverändert übernommen werden. Lediglich die Schnittstellen zu Metadaten-Schema und AAI wurden angepasst, bzw. neu entwickelt. Für den produktiven Betrieb werden an den beiden letztgenannten Schnittstellen noch Anpassungen nötig sein.

Weiterhin wurden umfangreiche Anpassungen im Sinne von Code-Generalisierung durchgeführt, so dass nun ein gemeinsamer Code-Stamm der beiden CRUD-Dienste existiert, mit verschiedenen Maven-Modulen für TG- und DH-crud, die gemeinsame Grundmodule nutzen.

[19] Vgl. koLibRI – kopal Library for Retrieval and Ingest. http://dp4lib.langzeitarchivierung.de/index_koLibRI.php.de

5.2.1. TextGrid-crud

TG-crud ist der Dienst, der für das TextGrid-Repository alle grundlegenden Speicheroperationen über eine SOAP- und REST-API bietet (CREATE, RETRIEVE, UPDATE und DELETE). Eine umfassende Dokumentation befindet sich im öffentlichen TextGrid-Wiki[20]. Alle diese Operationen können (je nach Verfügbarkeit) auf dem dynamischen TextGrid-Repository (hier wird mit den Daten gearbeitet, z. B. aus dem TextGridLab heraus) sowie auch auf dem statischen ausgeführt werden. Im statischen TextGridRep liegen alle publizierten Daten. Diese sind unveränderbar, mittels eines PID referenzierbar und weltweit lesbar.

TG-crud wird hauptsächlich vom TG-lab aus angesprochen, kann aber auch, sofern eine RBAC SessionID und ggf. eine TextGrid ProjektID vorhanden ist, von anderen Clients oder per Web-Browser genutzt werden.

Der TG-crud-Quellcode[21] ist bereits weitestgehend so modularisiert, dass es für die Anbindung an eine AAI, an einen Identifier sowie an diverse Storage-Knoten bereits Interfaces gibt, die für die verschiedenen Storage-Backends, Datenbanken bzw. Identifier- und AAI-Anbindungen nahezu beliebig implementiert werden können. Es existieren bereits Implementierungen für:

- Storage
  - JavaGAT[22] und Fedora[23] (Storage)
  - eXist[24] (XML-Datenbank)
  - ElasticSearch[25] (Index-Datenbank)
  - Sesame (RDF-Datenbank)
- AAI
  - TextGrid-RBAC[26] (Tgextra)
- Identifier
  - UUID
  - NOID[27]

---

[20] Vgl. TG-crud-Dokumentation im öffentlichen TextGrid-Wiki: https://dev2.dariah.eu/wiki/display/TextGrid/Main+Page

[21] Vgl. TG-crud Quellcode im GIT: https://projects.gwdg.de/projects/tg-crud/repository

[22] Vgl. JavaGAT Home. http://gforge.cs.vu.nl/gf/project/javagat/

[23] Vgl. Fedora Repository. http://www.fedora-commons.org/

[24] Vgl. eXistdb - The Open Source Native XML Database. http://www.exist-db.org/exist/apps/homepage/index.html

[25] Vgl. Elasticsearch: RESTful, Distributed Search & Analytics. https://www.elastic.co/products/elasticsearch

[26] Vgl. openRBAC project. http://openrbac.de/en_startup.xml

[27] Vgl. NOID: Nice Opaque Identifier (Minter and Name Resolver). https://wiki.ucop.edu/display/Curation/NOID

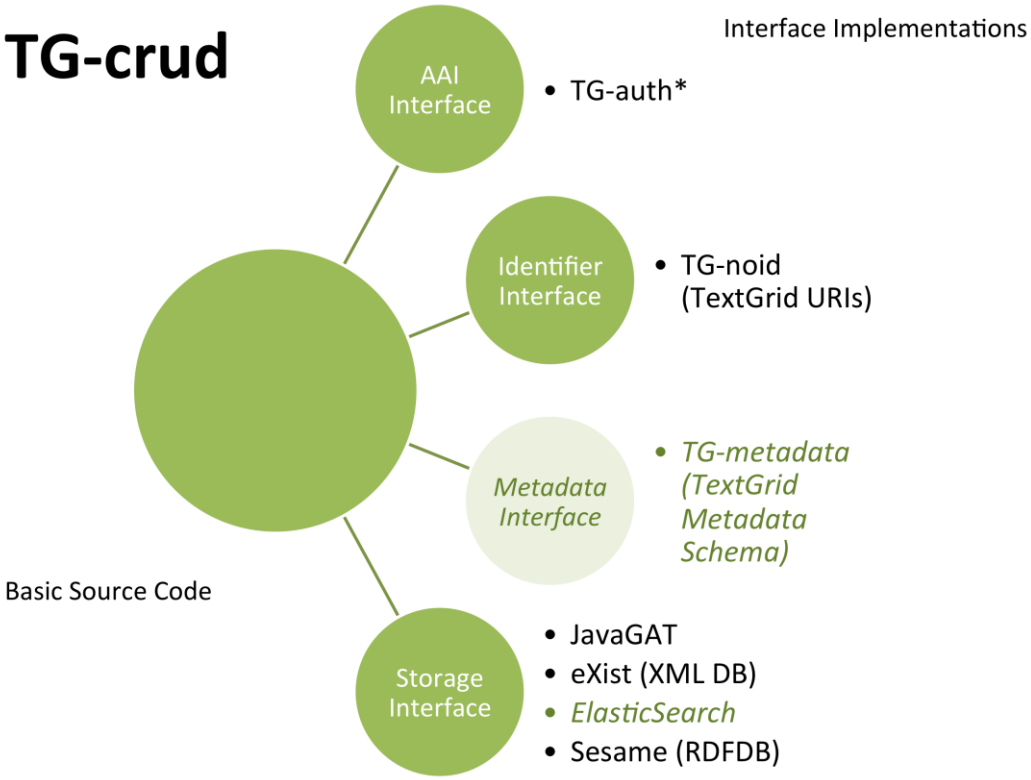

Abbildung 18: Schnittstellen des TG-crud Services

5.2.2. DARIAH-crud

Der DARIAH-crud führt grundlegende Operationen auf dem DARIAH-DE-Repositorium aus, zunächst also CREATE-, RETRIEVE-, UPDATE- und DELETE-Operationen. Im Gegensatz zum TextGrid-Repository soll das DARIAH-Repositorium jedoch nur als statisches Repositorium aufgebaut werden, für alle dynamischen Vorgänge ist der DARIAH-DE OwnStorage nutzbar. Über die Publish-GUI bzw. den Publish-Service ist zunächst nur CREATE möglich (interner Zugriff für interne Services), für READ- und READMETADATA-Operationen gibt es einen zusätzlichen DARIAH-crud, der nur Lese-Operationen erlaubt (öffentlicher Zugriff für alle Nutzerinnen und Nutzer).

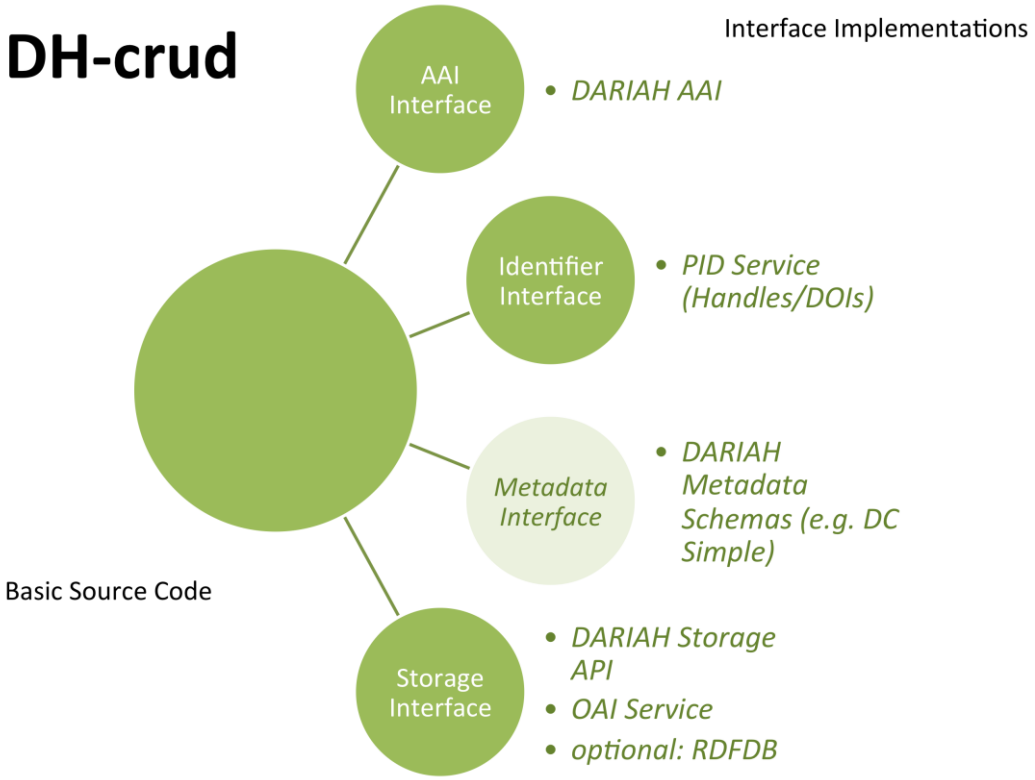

Abbildung 19: Schnittstellen des DH-crud Services

5.3. OAI-PMH

Der OAI-PMH-Service des DARIAH-DE-Repositoriums liefert die Daten für die Generische Suche von DARIAH-DE, so dass alle im Repositorium verfügbaren Inhalte such- und auffindbar sind. Dafür werden beim Einspielen der Daten per DARIAH-crud alle relevanten Metadaten in einen Index geschrieben, auf den der OAI-PMH Data Provider schnell zugreifen und dementsprechende OAI-Anfragen bedienen kann. Wie mit Volltexten umgegangen wird, ist noch zu klären, evtl. ist das OAI-PMH Protokoll hierfür weniger geeignet. Auf jeden Fall soll die Generische Suche auch Volltextsuche über die DARIAH-DE-Repositoriums-Inhalte ermöglichen. Sowohl die Generische Suche als auch der TextGrid Suchdienst TG-search nutzen eine ElasticSearch-Datenbank als Datenindex, beides ist bereits produktiv im Einsatz.

Ein OAI-PMH Data-Provider wurde bereits für das TextGrid-Repositorium implementiert.[28] Dieser wird in der Collection Registry eingetragen und für die Generische Suche genutzt.

[28] Vgl. OAI-PMH DARIAH/TextGrid im DARIAH-Wiki. https://dev2.dariah.eu/wiki/x/f4Q9AQ

5.4. PID Service

Der TextGrid PID-Service ist als Wrapper-Service implementiert, der als Vermittler zwischen den nutzenden Diensten (TG-crud/publish und DH-crud/publish) fungiert. Da nicht jeder sich PIDs erzeugen kann bzw. können soll, sondern dies nur mit einem Account bei der GWDG möglich ist, kapselt dieser Service zum Einen den Zugriff auf die Funktionen des Handle-Services der GWDG, und bedient außerdem einige besondere Anforderungen, die für die nutzenden TextGrid- und DARIAH-Services notwendig sind. Der TextGrid PID-Service wurde um einige DARIAH-spezifische Funktionen erweitert und kann nun so konfiguriert werden, dass ein Betrieb von zwei Instanzen des Services problemlos möglich ist.

5.5. Hochverfügbarkeit (High-Availibility) und Parallelisierung

Alle DARIAH-Dienste, die zum Kern des Repositoriums gehören, sollten durch ein HA-Konzept abgesichert sein. Dazu gehören – im Sinne der TextGrid HA – zunächst der DARIAH-crud, der OAI-PMH-Service sowie der Identifier-Service (z. B. der GWDG PID-Service). TG-publish ist zunächst nicht in den Kern des TextGrid HA-Konzept aufgenommen werden, da TextGrid (TextGridLab und TextGridRep mit allen Kernfunktionen) auch ohne TG-publish laufen. Ein Ausfall von TG-publish legt also nicht den kompletten TextGrid-Betrieb lahm. Bei dem DARIAH Publish Service wird das anders sein, da Publizieren schließlich die Hauptaufgabe des DARIAH Publish Services ist.

Aufbauend auf den Erfahrungen der TextGrid HA kann DARIAH eine Hochverfügbarkeit für die Kernkomponenten des DARIAH Repositoriums installieren. Folgende Dienste müssen auf die Möglichkeiten von Parallelisierung sowie Hochverfügbarkeit geprüft werden:

- Die DARIAH-crud HA kann direkt vom TextGrid-Konzept übernommen werden.
- OAI-PMH ist ebenfalls in TextGrid vorhanden, hier können ebenfalls die Maßnahmen von TextGrid übernommen werden.
- Der PID-Service (EPIC2) wird von der GWDG entwickelt und bereit gestellt, hier wird eine Hochverfügbarkeit zunächst vorausgesetzt.
- Der Publish Service ist in TextGrid zunächst nicht als Kernservice vorgesehen, dies sollte in DARIAH jedoch überlegt werden, da dieser Dienst hier eine sehr zentrale Rolle spielt.

6. Abbildungsverzeichnis