

Wiki » Feinkonzepte und Implementierungen »

2.1 Schnittstellen

OpenInfRA ist als Infrastruktur-Software geplant. Für den Austausch von Daten zwischen heterogenen Systemen ist die Bereitstellung von Schnittstellen eine der wesentlichen Voraussetzungen einer Infrastruktur-Software, welche sich in eine bestehende Systemlandschaft integrieren soll. Zusätzlich kann nur über die Bereitstellung von Schnittstellen eine langfristige Nutzbarkeit der gespeicherten und aufbereiteten Daten erreicht werden.

In der folgenden Tabelle werden die insgesamt 19 Anforderungen zu Schnittstellen, die im Anforderungskatalog festgehalten sind, aufgelistet. Die Anforderungen, die Grau unterlegt sind, werden durch den Arbeitnehmer (Delphi IMM) umgesetzt und in diesem Kapitel nicht weiter betrachtet. Die anderen Anforderungen werden im Folgenden näher erläutert.

ID	Priorität	Thema	Anforderung	Beschreibung	Umsetzung durch AN	Umsetzung durch AG
IFS_0070	KO	Bidirektionale Datei-Schnittstellen	XML	Das System soll eine bidirektionale XML-Schnittstelle bereitstellen.	Wird realisiert.	-
IFS_0080	KO	Bidirektionale Datei-Schnittstellen	CSV	Das System soll eine bidirektionale CSV-Schnittstelle bereitstellen.	Wird realisiert.	-
IFS_0090	KO	Bidirektionale Datei-Schnittstellen	Schnittstelle für Bilddaten	Das System soll eine bidirektionale Schnittstelle für Bilddaten (u.a. JPG, BMP, PNG, TIFF) bereitstellen und EXIF-Metadaten berücksichtigen.	Wird realisiert.	-
IFS_0100	3	Bidirektionale Datei-Schnittstellen	Schnittstelle für Listen/Tabellen	Das System soll eine bidirektionale Schnittstelle für Listen/Tabellen im z.B. CSV (s.auch IFS_0080), XLS, ods oder tab-Format bereitstellen.	Realisierung: für CSV	-
IFS_0160	3	Bidirektionale Datei-Schnittstellen	XMP	Das System soll eine bidirektionale XMP-Schnittstelle bereitstellen.	Wird nicht realisiert.	Zur Zeit keine Realisierung.
IFS_0180	N	Bidirektionale Datei-Schnittstellen	SQL	Das System soll eine bidirektionale SQL-Schnittstelle bereitstellen.	Wird nicht realisiert.	Abgewiesen
IFS_0120	1	Bidirektionale Schnittstellen zu Desktop-GIS	SHP	Das System soll eine bidirektionale Schnittstelle für Shape-Dateien (SHP) bereitstellen.	Wird realisiert.	-
IFS_0130	2	Bidirektionale Schnittstellen zu Desktop-GIS	GML	Das System soll eine bidirektionale Schnittstelle für GML bereitstellen.	Wird realisiert.	-
IFS_0140	2	Bidirektionale Schnittstellen zu Desktop-GIS	KML	Das System soll eine bidirektionale Schnittstelle für KML bereitstellen.	Wird realisiert.	-
IFS_0150	3	Bidirektionale Schnittstellen zu Desktop-GIS	AutoCAD-Formate	Das System soll eine bidirektionale Schnittstelle für AutoCAD-Formate (insbesondere DXF, DWG) bereitstellen.	Wird nicht realisiert.	Zur Zeit keine Realisierung.
IFS_0190	KO	Druck-Schnittstelle	PDF	Das System soll eine unidirektionale PDF-Schnittstelle für den konfigurierbaren Druck-Export von Formularen bereitstellen.	Wird realisiert.	-
IFS_0200	N	Schnittstelle externer Zugriff	Schnittstelle für externen Zugriff auf OpenInfRA	Das System soll eine Schnittstelle für den Zugriff externer Systeme auf OpenInfRA-Inhalte bereitstellen.	Wird nicht realisiert.	Realisierung einer REST-Schnittstelle.
IFS_0210	KO	Schnittstelle Rollen-Rechte-System	Schnittstelle für Rollen-Rechte-System	Das System soll eine Rollen- und Rechte-Schnittstelle bereitstellen, die Benutzerrollen und -rechte für die Verwendung in Modulen der OpenInfRA-Applikation regelt. Z.B.: Datenbank, Suche, WebGIS	Wird realisiert.	-
IFS_0010	N	Service-Schnittstellen	Lesender Zugriff auf das DAI-interne System: Bibliothekskatalog ZENON	Das System soll eine Schnittstelle für den lesenden Zugriff auf den Bibliothekskatalog ZENON bereitstellen.	Wird nicht realisiert.	Wird realisiert.
IFS_0020	N	Service-Schnittstellen	Lesender Zugriff auf das DAI-interne System: Gazetteer	Das System soll eine Schnittstelle für den lesenden Zugriff auf einen Gazetteer bereitstellen.	Wird nicht realisiert.	Wird realisiert.
IFS_0040	N	Service-Schnittstellen	Lesender und schreibender Zugriff auf: Bildarchiv Arachne	Das System soll eine Schnittstelle für den lesenden und schreibenden Zugriff auf das Bildarchiv Arachne bereitstellen.	Wird nicht realisiert.	Wird realisiert.
IFS_0050	N	Service-Schnittstellen	Lesender Zugriff auf das DAI-interne System: Archive	Das System soll eine Schnittstelle für den lesenden Zugriff auf Grabungsarchive bereitstellen.	Wird nicht realisiert.	Abgewiesen
IFS_0060	N	Service-Schnittstellen	Lesender Zugriff auf das DAI-interne System: Stichwerkbrowser (Köln / Heidelberg)	Das System soll eine Schnittstelle für den lesenden Zugriff auf den Stichwerkbrowser (Köln / Heidelberg) bereitstellen.	Wird nicht realisiert.	Abgewiesen
IFS_0170	N	Service-Schnittstellen	Lesender Zugriff auf das Online-System: http://pleiades.stoa.org/places	Das System soll eine Schnittstelle für den lesenden Zugriff auf http://pleiades.stoa.org/places bereitstellen.	Wird nicht realisiert.	Wird realisiert.

2.1.1 Bidirektionale Datei-Schnittstellen: XMP

Die Extensible Metadata Platform (XMP) (<http://www.adobe.com/products/xmp/overview.html>) ist ein von Adobe zertifizierter Standard zur Integration von Metadaten in Bilddateien. Diese Metadaten beziehen sich z.B. auf folgende Informationen: Name des Fotografen, Belichtungszeit, verwendete Ausrüstung oder Aufnahmeort. Wie auf der Webseite von Adobe beschrieben, ist XMP ein Standard, der eine automatische Verarbeitung von Bilddaten ermöglichen soll und Funktionen des semantischen Webs mit einbezieht.

Adobe stellt für XMP ein SDK bereit (<http://www.adobe.com/devnet/xmp.html>), welches quelloffen, unter der **BSD-Lizenz** veröffentlicht ist und in C++ zur Verfügung steht. Zusätzlich wird auch eine Java-Bibliothek angeboten. Aufgrund der bereitgestellten Java-Bibliothek kann das XMP-Dateiformat in OpenInfRA integriert werden.

Aufgrund der geringen Priorität dieser Anforderung (Priorität 3) im Anforderungskatalog wird diese Anforderung zunächst nicht umgesetzt und hier nicht weiter beschrieben.

Offene Fragen: Wie genau soll diese bidirektionale Schnittstelle aussehen bzw. welcher Funktionsumfang ist gewünscht? Z.B. ist es möglich, dass beim Hochladen einer XMP-Datei in OpenInfRA die enthaltenen Metadaten angezeigt werden. Zusätzlich können diese Metadaten in der OpenInfRA-Datenbank (redundant) gespeichert werden. Vielleicht ist damit aber auch gemeint, dass wenn ein XMP-Bild in OpenInfRA betrachtet wird, dass die enthaltenen Metadaten entsprechend angezeigt werden. Mit bidirektional ist sicherlich nicht nur der Import (Hochladen) sondern auch der Export (runterladen) gemeint. Wie soll das verstanden werden? Sollen einfache Bilder (PNG, JPEG) im XMP-Format konvertiert werden können vor dem Export oder sollen bereits existierende XMP-Bilder exportiert werden. Sollen die Metadaten von XMP-Bildern in OpenInfRA bearbeitet werden können?

2.1.2 Bidirektionale Datei-Schnittstellen: SQL

Diese Anforderung wird abgewiesen. Dafür gibt es folgende Gründe:

1. OpenInfRA wird ein sehr umfangreiches Benutzer/Rechte/Rollensystem bekommen. Durch die Bereitstellung einer SQL-Schnittstelle wäre dieses Benutzer/Rechte/Rollensystem außer Kraft gesetzt.
2. Die verwendeten Datenbankschemata sowie die dazugehörigen Integritätsbedingungen sind äußerst komplex. Aus diesem Grund setzen direkte SQL-Anfragen an das OpenInfRA-Datenmodell ein detailliertes Expertenwissen durch den Benutzer voraus.
3. Das Bereitstellen einer direkten SQL-Schnittstelle birgt ein erhebliches Sicherheitsrisiko.
4. Das verwendete DBMS (Postgres) enthält bereits eine SQL-Schnittstelle. Nach Abwägung des Sicherheitsrisikos kann die bereits vorhandene SQL-Schnittstelle für einen ausgewählten Benutzerkreis freigegeben werden. Dies ist zwar möglich wird aber hier ausdrücklich nicht empfohlen!

2.1.3 Bidirektionale Schnittstellen zu Desktop-GIS: AutoCAD-Formate

Eine Realisierung ist durchaus denkbar. Insbesondere die [Drawing Interchange Format \(DXF\)-Schnittstelle](#) als quelloffene Schnittstelle ist für die Umsetzung in OpenInfRA geeignet.

Aufgrund der geringen Priorität dieser Anforderung (Priorität 3) im Anforderungskatalog wird diese Anforderung zunächst nicht umgesetzt und hier nicht weiter beschrieben.

Offene Fragen: Welche Dokumente/Zeichnungen/Bilder sollen in AutoCAD-Formate exportiert werden?

2.1.4 Schnittstelle externer Zugriff: REST

Representational State Transfer (REST) ist ein Programmierparadigma für Web-Anwendungen.

2.1.5 Service-Schnittstellen

Wie in der folgenden Abbildung dargestellt, gibt es drei Möglichkeiten für die Umsetzung der Service-Schnittstellen. Die erste Möglichkeit basiert auf dem einfachen Kopieren einer URI im Zielsystem und dem Einfügen der kopierten URI in OpenInfRA. Die zweite Möglichkeit basiert auf einem direkten Zugriff auf die (wenn vorhanden) REST-Schnittstellen mittels *Asynchronous JavaScript and XML (AJAX)* und *JavaScript Object Notation (JSON)*. Die dritte Möglichkeit basiert auf einem Proxy (Vermittler). Diese drei Möglichkeiten wurden zunächst prototypisch umgesetzt, um die Vor- bzw. Nachteile der drei Lösungen im Detail betrachten zu können. Im Folgenden werden diese drei Möglichkeiten genauer beschrieben und die Vor- bzw. Nachteile miteinander verglichen.

Kopieren/Einfügen

Dies ist die einfachste und universellste Methode, die auch von DAI vorgeschlagen wurde. Dazu kann die bereits vorhandene Suche im Zielsystem verwendet werden. Der Link zu den gewünschten Inhalten kann dann einfach kopiert und in OpenInfRA eingefügt werden. Diese Methode enthält folgende Vorteile:

- minimaler Entwicklungsaufwand (die bestehende Suche und deren Logik kann genutzt werden)
- diese Methode ist äußerst leichtgewichtig (keine zusätzliche Rechenlast auf dem Server oder Client)

Diese Methode enthält die folgenden Nachteile:

- nicht besonders nutzerfreundlich (der Benutzer muss die OpenInfRA-Webseite verlassen)
- der gewünschte Link muss kopiert und eingefügt werden (evtl. falsche Links müssen erkannt werden)

Fazit: Die Methode ist einfach umsetzbar und wird auch als Standard mit angeboten werden.

AJAX/JSON

Bei dieser Methode wird die vorhandene REST-Schnittstelle dazu genutzt, um Inhalte und Informationen des Zielsystems dynamisch in OpenInfRA darzustellen. Eine Voraussetzung dieser Methode ist, dass das Zielsystem eine entsprechende REST- oder AJAX-Schnittstelle anbietet. Dazu kann der Nutzer Suchanfragen in OpenInfRA formulieren, die dann an das Zielsystem mittels AJAX weitergeleitet werden. Als Antwort liefert das Zielsystem JSON-Objekte, die dann dynamisch während der Laufzeit in der OpenInfRA-Oberfläche integriert werden. Diese Methode hat die folgenden Vorteile:

- besonders nutzerfreundlich (der Benutzer muss OpenInfRA nicht verlassen, um externe Inhalte/Informationen zu kombinieren)
- diese Methode ist leichtgewichtig (keine zusätzliche Rechenlast auf dem Server; vertretbarer zusätzlicher Aufwand auf dem Client für AJAX-Anfragen)

Diese Methode enthält die folgenden Nachteile:

- relativ hoher Entwicklungsaufwand (bereits existierende Suchfunktionalitäten/Logiken des Zielsystems müssen erneut für OpenInfRA implementiert werden)
- bei Änderungen der REST-Schnittstelle des Zielsystems muss auch diese Implementierung angepasst werden

Fazit: Diese Methode ist relativ einfach umsetzbar. Um den Entwicklungsaufwand zu reduzieren wird nur eine minimale Suchfunktionalität (z.B. Schlagwortsuche) umgesetzt, die nicht die gesamte Komplexität des Zielsystems abdeckt. Hier ist insbesondere eine Kombination der ersten und der zweiten Methode denkbar. Für die einfache Suche kann das AJAX/JSON verwendet werden. Ist die bereitgestellte Suchfunktionalität für den Anwendungsfall zu gering, dann kann das Zielsystem aufgesucht werden. Dieses bietet dann die erweiterte Suchfunktionalität an.

Proxy

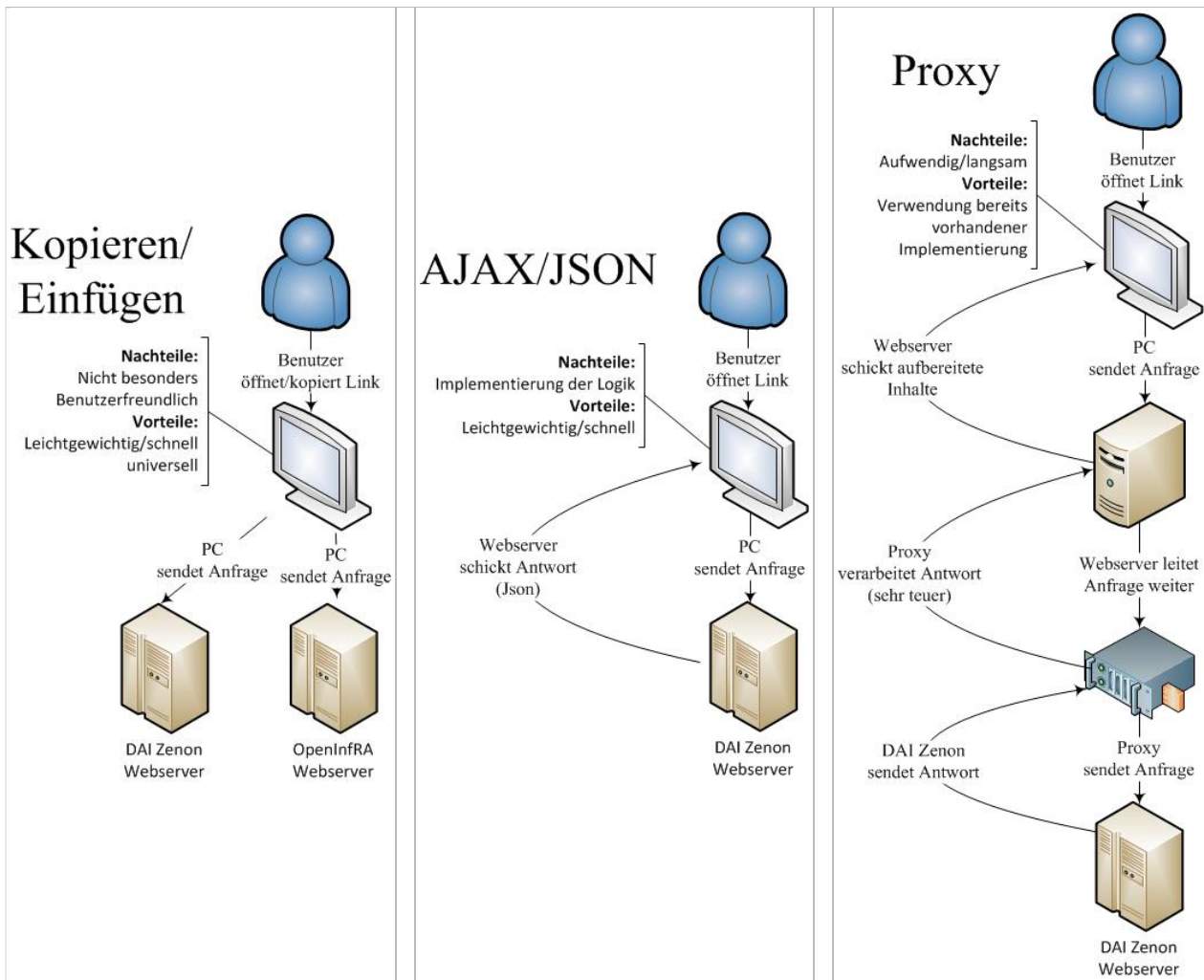
Die Idee der letzten Methode bestand darin, dass die Suchfunktionalität des Zielsystems in OpenInfRA nicht hätte nachempfunden werden müssen wenn das Zielsystem in OpenInfRA integriert werden könnte. Als Möglichkeit dazu bietet sich z.B. ein *Iframe* oder das Nachladen von einzelnen Inhalten in OpenInfRA mittels AJAX an. Seit HTML5 wurden allerdings die Richtlinien für *Cross-Origin Resource Sharing (CORS)* eingeführt. Somit ist es nicht möglich diese Methode ohne den Einsatz weiterer Modifikationen einzusetzen. Als eine mögliche Modifikation kann hier ein Proxy (Vermittler) angesehen werden. Ruft der Benutzer eine Webseite auf, so wird dieser Aufruf an einen Proxy weitergeleitet. Der Proxy ruft dann die gewünschten Web-Inhalte auf und verarbeitet diese. Somit können die CORS-Richtlinien effektiv umgangen werden. Leider stellte sich heraus, dass der Entwicklungsaufwand für den Proxy relativ hoch ist. Zudem verbraucht die interne Verarbeitung der Web-Inhalte zusätzliche Rechenressourcen wodurch sich die Antwortzeiten deutlich erhöhen. Diese Methode hat die folgenden Vorteile:

- es kann die originale Suchfunktionalität verwendet werden (Änderungen der Suchfunktionalität sind automatisch mit enthalten)
- besonders nutzerfreundlich (der Benutzer muss OpenInfRA nicht verlassen, um externe Inhalte/Informationen zu kombinieren)

Diese Methode enthält die folgenden Nachteile:

- relativ hoher Entwicklungsaufwand (es könnte aber auch bereits existierender Java-Proxy verwendet werden)
- relativ hohe Latenzzeiten (schwergewichtig)
- Sicherheitsrisiko: Der Proxy könnte dazu verwendet werden, um Webseiten mit fraglichen Inhalten aufzurufen. Dabei würde der Anfragerechner gegenüber dem Zielsystem unerkannt bleiben, denn der Proxy leitet diese Anfrage weiter. (Einschränkung durch 'whitelist' möglich)

Fazit: Rein subjektiv betrachtet ist diese Methode am benutzerfreundlichsten und die bereits beim Zielsystem existierende Suchfunktionalität/Logik muss nicht nachempfunden werden. Dennoch ist diese Methode gegenüber den ersten beiden Methoden relativ aufwendig in der Entwicklung. Zusätzlich werden relativ viele Rechenressourcen benötigt was sich negativ auf die Latenzzeiten auswirken kann. Wie auch gemeinsam bei einem OpenInfRA-Projekttreffen entschieden, wird diese Methode nicht weiter verfolgt.



2.1.6 Umsetzung der Service-Schnittstellen

In diesem Kapitel wird die geplante Umsetzung der Service-Schnittstellen beschrieben. Zu Beginn war geplant, dass für alle Service-Schnittstellen eine Bibliothek erstellt wird, welche dann die Funktionalitäten bündelt. Diese anfängliche Planung wurde aus den folgenden Gründen verworfen.

1. Zunächst ist aufgefallen, dass insbesondere die DAI-Systeme sehr unterschiedlich sind. Aufrufe, die die von den DAI-Systemen angebotenen Funktionalitäten aufrufen, müssen für jedes DAI-System individuell angepasst sein. Aus diesem Grund ist das Zusammenfassen in einer Bibliothek grundsätzlich möglich. Inhaltlich bleiben die Funktionsaufrufe trotzdem stark voneinander separat. Somit ergibt sich eine relativ große Bibliothek mit nur einer geringen Anzahl von wiederverwendbarem Quellcode.
2. Durch die Zerlegung der Service-Schnittstellen in mehrere dedizierte Bibliotheken ergibt sich der Vorteil, dass in OpenInfRA je nach Bedarf einzelne Erweiterungen geladen werden können. Somit kann eine Erweiterung ausgetauscht oder sogar ausgelassen werden, ohne dass Seiteneffekte auf andere Service-Schnittstellen auftreten. Dies kann z.B. notwendig sein, wenn sich die Schnittstellen der Zielsysteme ändern.

Für die Umsetzung der Service-Schnittstellen werden die folgenden externen Bibliotheken verwendet:

- jQuery (<http://jquery.com/>) in der Version 1.11.1 (Kompatibilität mit älteren IE Browsern)
- tablesorter (<http://tablesorter.com/docs/>) 2.0.5 (derzeit geplant)

Bibliothekskatalog ZENON

Der Bibliothekskatalog ZENON kann unter der Adresse <http://zenon.dainst.org/> aufgerufen werden. Zum Aufruf der Suchfunktion mittels REST-Schnittstelle steht derzeit folgende URI zur Verfügung:

- 'http://zenon.dainst.org:8080/elwms-zenon/search?q='.

Der Aufruf einer einzelnen Zenon-Instanz (Literatur-Instanz) lautet:

- <http://zenon.dainst.org:8080/elwms-zenon/resource/> + zenonId + ?format=standard

Je nach Änderung am ZENON-System können die oben genannten URI's verändert werden und müssen dann entsprechend auch in OpenInfRA angepasst werden.

Achtung: Der Aufruf der Rest-Schnittstelle von Zenon ist immer über eine URI-Encoding durchzuführen (JavaScript-Methode: `encodeURIComponent(uri)`). Nur so kann auch nach Umlauten und Sonderzeichen gesucht werden.

Für die Implementierung der Service-Schnittstelle ZENON sind die folgenden Teile umgesetzt:

1. Für die Umsetzung der Suchfunktionalität gibt es zunächst eine Überschrift, die auf das ZENON-System verweist (optische Erkennung, dass es sich um eine Suche im ZENON-System handelt). Beim Klicken auf diese Überschrift wird die Zenon-Webseite in einem neuen Browser-Fenster geöffnet. Die Zenon-Webseite ermöglicht die erweiterte bzw. detaillierte Suche im Zenon-System.
2. Weiterhin bietet die Implementierung ein Suchfeld an. Dieses Suchfeld ermöglicht eine einfache Suche mit folgenden Eingabemöglichkeiten: Freitext, Systemnummer, ISBN oder ZENON-URI. Einige der Literatureinträge sind einem Gazetteer-Schlüssel (Ort) zugeordnet. Aus dem Grund gibt es zusätzlich zu den genannten Eingabemöglichkeiten auch noch die Möglichkeit einen Gazetteer-Schlüssel als Suchparameter einzugeben.
3. Die Ergebnisse der Suche werden in einer tabellarischen Ansicht dargestellt. Dazu gehören die folgenden Informationen: Typ der Ressource (z.B. eBook, Buch), Imprint, Autoren und Titel.
4. Zusätzlich enthält jedes Suchergebnis zwei Buttons. Der erste Button kann dazu verwendet werden, um die entsprechende ZENON-Seite mit der ausgewählten Literatur-Referenz zu öffnen. Der zweite Button kann dazu verwendet werden, um die ausgewählte Literatur-Referenz zu speichern.
5. Die Umsetzung der Mehrsprachigkeit bleibt zunächst offen.

Aufgrund wiederholter Ausfälle des ZENON-Servers musste die Entwicklung zeitweise ausgesetzt werden.

DAI Gazetteer

Der Gazetteer kann unter der Adresse <https://gazetteer.dainst.org> aufgerufen werden. Zum Aufruf der Suchfunktion mittels REST-Schnittstelle steht derzeit folgende URI zur Verfügung:

- 'https://gazetteer.dainst.org/search.json?q='.

Achtung: Der Aufruf der Rest-Schnittstelle ist immer über eine URI-Encoding durchzuführen. Nur so kann auch nach Umlauten und Sonderzeichen gesucht werden. Die entsprechende JavaScript-Methode heißt: `encodeURIComponent(uri)`.

Für die Implementierung der Service-Schnittstelle DAI Gezatteeer sind die folgenden Teile geplant:

1. Für die Umsetzung der Suchfunktionalität gibt es zunächst eine Überschrift, die auf das Gazetteer-System verweist (optische Erkennung, dass es sich um Gazetteer handelt).

Bildarchiv Arachne

Das Bildarchiv Arachne kann unter der Adresse <http://arachne.uni-koeln.de/drupal/> aufgerufen werden.


Archive

Basierend auf einer Rücksprache mit dem DAI wird diese Anforderung bzw. die Service-Schnittstelle zum DAI-internen System: Archive nicht umgesetzt. Dazu hat das DAI folgende Aussage getroffen: "Ein Archivsystem hat es nie gegeben".


Stichwerkbrowser

Basierend auf einer Rücksprache mit dem DAI wird diese Anforderung bzw. die Service-Schnittstelle zum DAI-internen System: Stichwerkbrowser nicht umgesetzt. Dazu hat das DAI folgende Aussage getroffen: "das Stichwerkprojekt ist zu einer Unterprojekt von Arachne geworden".

[proxy.png](#) (52,919 KB)  Tino Noack, 14.08.2014 08:55

[json.png](#) (34,098 KB)  Tino Noack, 14.08.2014 09:13

[copy.png](#) (37,938 KB)  Tino Noack, 14.08.2014 09:22

[copy.jpg](#) (47,118 KB)  Tino Noack, 21.10.2015 14:26