

Softwarearchitektur für semantische Stadtteilplattformen

HTWK Leipzig, Studiengang Master Medieninformatik (M.Sc.)
Tobias Hahn

29.02.2016

1 Einleitung

In der vorliegenden Arbeit entsteht ein Architekturkonzept zur Erstellung einer stadtteilübergreifenden semantischen Plattform für Leipzig. Es wird anhand der Plattform *Leipziger Ecken* diskutiert, was eine Stadtteilplattform ist und wie diese auf weitere Stadtteile ausgeweitet werden kann. Besonderer Fokus wird dabei auf die Bereitstellung der semantisch relevanten Daten gelegt. Diese sollen in, im Semantic Web nutzbare, Dateiformate transformiert und auf möglichst weit verbreitete Vokabulare abgebildet werden. Dabei sollen die Betreiber der einzelnen Stadtteilplattformen jedoch möglichst wenig Verantwortung für die semantische Aufbereitung der Daten haben.

Der Architekturentwurf wird dabei möglichst abstrakt behandelt. Somit entsteht ein allgemeingültiger Entwurf, der auf andere Domänen abgebildet werden kann. Es findet darüber hinaus eine Einordnung im *Helois Common Research Model* statt.

Darauf basierend findet eine Identifikation von Komponentenkategorien statt, die in ihrer Gesamtheit die allgemeine Softwarearchitektur, und damit die stadtteilübergreifende semantische Plattform bilden. Es werden der Funktionsumfang und die Verantwortlichkeiten der einzelnen Komponentenkategorien im Hinblick auf den soziokulturellen Hintergrund dargestellt.

Da nicht auf jede Facette einer Softwarearchitektur, bspw. dem detaillierten Datenkonzept und einem genauen Betreiberkonzept, eingegangen werden kann, werden am Ende der Arbeit einige Ansatzpunkte für zukünftige weitere Arbeiten genannt.

2 Rahmenbedingungen

2.1 Leipziger Ecken

2.1.1 Beschreibung

Leipziger Ecken ist ein Onlineportal. Das Portal bezeichnet sich selbst als "... eine interaktive Stadtteilplattform für die Bezirke des Leipziger Ostens, getragen und verwaltet vom "Helden wider Willen e.V.", Hildegardstraße 49, 04315 Leipzig"[wWe]. Fortlaufend wird das Portal als Stadtteilplattform

bezeichnet. Ziel der Plattform ist es, den öffentlichen Dialog zu fördern. *Akteure* sollen dabei ihre *Events* und *Orte* einstellen können, sodass sie für andere Akteure sowie einfache Nutzer zur Verfügung stehen (Eine genaue Definition der hier genannten Entitäten erfolgt in 2.1.3).

Die Stadtteilplattform bezieht sich zum Zeitpunkt der Erstellung der vorliegenden Arbeit ausschließlich auf den Leipziger Osten¹. Die Reichweite soll jedoch auf den gesamten Leipziger Raum ausgeweitet werden. Langfristig ist durch die Teilnahme an EU-Programmen eine Ausweitung auf EU-Ebene angestrebt.

2.1.2 Funktionsumfang

Die Stadtteilplattform hat stellt mehrere Funktionen für ihre Nutzer bereit. Darunter sind folgende essenzielle Funktionen.

- Kurzdarstellung der *Akteure* vor Ort (bspw. Organisationen/Vereine, Personen, etc.)
- Zugangsmöglichkeit für *Akteure*, um selbstständig Inhalte einzustellen und zu pflegen (Adresse, Ansprechpartner, Kontakt, Zuordnung der Organisation zu inhaltlichen Themenfeldern und Zielgruppen, Termine, Ressourcen etc.)
- Darstellung von *Events*

Die Stadtteilplattform bietet darüber hinaus weitere Funktionen, die für diese Arbeit jedoch keine Relevanz haben. Darunter befindet sich zum Beispiel ein gemeinsamer Blog und die Vernetzungsmöglichkeit einzelner Nutzer der Website. Diese Funktionen finden hier keine Beachtung.

2.1.3 Entitäten der Stadtteilplattformen

Für die vorliegende Arbeit wird nur ein Teil der gesamten Stadtteilplattformen betrachtet. Dies bezieht sich auf den Funktionsumfang sowie die für den Inhalt dieser Arbeit wichtigen Entitäten.

Entitäten seien dabei als **technische** Datensätze definiert.

Sie sind innerhalb einer relationalen Datenbank persistiert. Jede Entität wird somit mit Hilfe einer Tabelle repräsentiert. Dies bedeutet jedoch nicht, dass jede Tabelle eine Entität repräsentiert. N:M-Relationen zwischen Entitäten werden in relationalen Datenbank Management Systemen (RDBMS) ebenfalls durch Tabellen abgebildet [War07, S. 111 ff].

Weiterhin hat nicht jedes Datum innerhalb einer Entität eine direkte semantische Bedeutung. So werden Relationen zu anderen Entitäten über deren (Fremd-) Schlüssel innerhalb der Entitätstabelle referenziert. Die semantischen Informationen findet man jedoch innerhalb der referenzierten Tabelle. Die für die Arbeit wichtigen Entitäten sind definiert als

- *Akteure*
Veranstalter, Vereine, Initiativen und Privatpersonen, die im Leipziger Osten auf verschiedene Art aktiv sind. Sie werden von den Nutzern der Stadtteilplattform gepflegt
- *Events*
Öffentlichen Veranstaltungen, die von einem Akteur organisiert werden.
- *Adressen*
Adressdaten der Akteure oder der Veranstaltungen.

Zwischen den Entitäten bestehen dabei die in der Tabelle 1 definierten Relationen.

¹Einzusehen unter <https://leipziger-ecken.de/leipziger-osten>

<i>Akteur - Event</i>	1 : m
<i>Aktuer - Adresse</i>	1 : 1
<i>Event - Adresse</i>	1 : 1

Table 1: Relationen zwischen den einzelnen Entitäten einer Stadtteilplattform

2.2 Semantische Stadtteilplattformen

Ziel ist es, die Stadtteilplattform auf weitere Stadtteile auszuweiten 2.1.1. So sollen mittelfristig alle Stadtteile vertreten sein. Dabei wird eine dezentrale Struktur angestrebt. Jeder Stadtteil soll für seine eigene Stadtteilplattform verantwortlich sein.

Es soll ein effektiver Austausch der stadtteilübergreifend relevanten Daten geschehen können. Das bedeutet, dass die Entitäten so angeboten werden müssen, dass möglichst viele Plattformen damit umgehen können. Dabei soll die Semantik der Daten im Vordergrund stehen, anstatt technische Abhängigkeiten zu erzeugen.

Die Daten der einzelnen Stadtteilplattformen sollen dem Semantic Web bereit gestellt werden.

2.3 Soziokultureller Hintergrund

Die Stadtteilplattformen agieren vor Allem auf kultureller und lokaler Ebene. Plattformbetreiber sowie Nutzer sollen auf Augenhöhe agieren und lokale Akteure sein, was die Dezentralität der einzelnen Plattformen unterstützt. Es gilt dabei zu beachten, dass ein Großteil der Mitarbeiter der einzelnen Stadtteilplattformen ehrenamtlich arbeiten. Somit kann bei den Mitarbeitern nicht von professionell geschulten IT-Mitarbeitern ausgegangen werden. Die Nutzer der Plattform sollen gleichermaßen zu der Entwicklung der Stadtteilplattformen beitragen. Sie sollen sie einerseits mit Leben füllen und andererseits die Entwicklung der Stadtteilplattform lenken. Dabei soll jeder Nutzer dazu in der Lage sein als Akteur zu fungieren und eigene Veranstaltungen einzustellen. Die Größe der Events ist dabei nicht relevant. So entsteht für jede Stadtteilplattform ein eigener Charakter, der sich von den anderen Stadtteilplattformen abheben wird. Dieser spiegelt dabei den Charakter des Stadtteils selbst wieder.

3 Ergebnisse der Seminararbeit

3.1 Einordnung der semantischen Stadtteilplattform

Der in Kapitel 2.2 eingeführte Begriff **Semantische Stadtteilplattform** soll nachfolgend verdeutlicht werden.

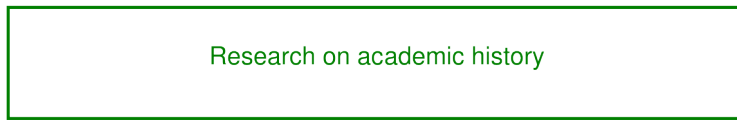
Der Begriff impliziert, dass die Daten der Stadtteilplattformen im **Semantic Web** bereitgestellt wird. Die Bereitstellung solcher Daten ist bisher auf vielerle Weise durchführbar [PB06, S. 73ff].

Um ein allgemeines und domänenübergreifendes Vorgehen entstehen zu lassen, wurde deshalb das **Heloise Common Research Model** erstellt [Rie]. Dabei handelt es sich um ein abstraktes Modell, was eine Schichtenarchitektur definiert, an derer man sich bei der Erstellung sowie dem Konsumieren von semantischen Daten orientieren kann.

Die Grafik 1 zeigt das Schichtenmodell. Man kann 3 Schichten erkennen. Nachfolgend werden diese von unten nach oben beschrieben.

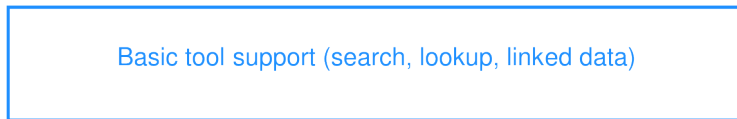
Das Repository Layer beschreibt domänenspezifische Datenbasen. Diese sind technisch nicht spezifiziert. Domänenspezifisch bedeutet dabei, dass die Daten für den jeweiligen fachlichen Kontext und

Research Interface Layer



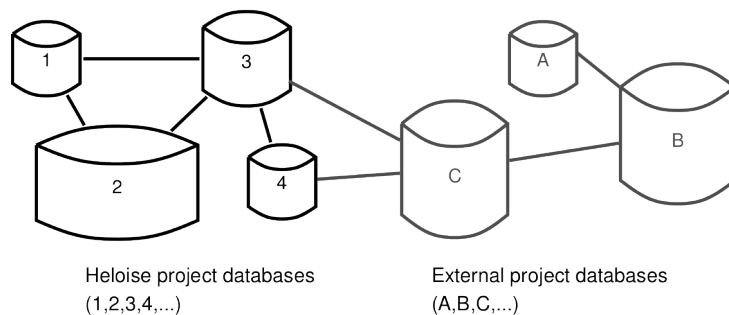
Domain specific meta-vocabulary
for historical research

Application Layer



Standardized vocabularies to support
access and interlinking on databases

Repository Layer



Domain specific research databases

Figure 1: Das HCRM Schichtenmodell

an die jeweilige technische Umsetzung angepasst und voneinander isoliert sind. Die einzelnen Datenbanken können somit zwar **semantisch** zusammenhängen, sind jedoch **technisch** unabhängig.

Das Application Layer bezeichnet dabei Tools, die das Konsumieren der Daten des Repository Layers ermöglichen. Diese Tools haben Zugriff auf die Daten und stellen diese in einer Art zur Verfügung, sodass sie von anderen Tools genutzt und interpretiert werden können.

Das Research Interface Layer nutzt die im Application Layer bereitgestellten Daten. Es definiert eigene Datensätze, die diese Daten domänenübergreifend aufbereitet und auswertet.

3.2 Stadtteilübergreifende Plattformen

3.2.1 Anforderungen der Stadtteilübergreifenden Plattform

Die in Kapitel 2.1.2 genannten Anforderungen beziehen sich vorrangig auf einen Stadtteil, hierbei dem Leipziger Osten².

Diesem sollen weitere Stadtteile hinzukommen, sodass mittelfristig alle Stadtteile vertreten sein können. Dabei ist wichtig, dass Synergieeffekte zwischen den einzelnen *Akteuren* ausgemacht und koordiniert werden können. Dafür sollen die von den *Akteuren* gepflegten Daten in einer Form hinterlegt werden, die es ermöglicht effektive **semantische Verbindungen** zu erstellen.

Die Stadtteilplattformen sollen dabei isoliert voneinander existieren können. So können die einzelnen Akteure ihre Daten unabhängig pflegen und jeder Stadtteil flexibel mit weiteren *Aktueren* zusammenarbeiten.

Der Datenaustausch soll ohne Aufwand seitens des Akteurs oder Nutzers stattfinden. Er muss jedoch

²Zu finden unter <http://leipziger-ecken.de/leipziger-osten>

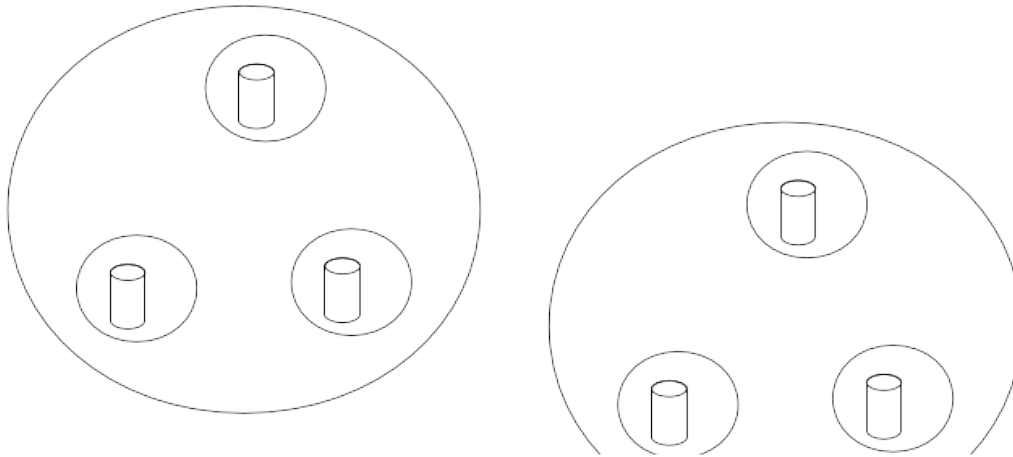


Figure 2: Einzelne isolierte Stadtteilplattformen

entscheiden können, welche Daten eingepflegt und öffentlich zur Verfügung gestellt sind.

3.2.2 Lokale Instanzen der Stadtteilplattform

Um weitere Stadtteile einzubeziehen, müssen auch diese eine Stadtteilplattform erstellen. Jede Stadtteilplattform basiert auf einer **einheitlichen Codebasis**, die gemeinsam gepflegt wird³. Grundsätzlich setzt die Stadtteilplattform auf dem Content Management System Drupal⁴ auf. Eine lokale Instanz der Stadtteilplattform bezeichnet somit eine instanziierte Stadtteilplattform. Sie ist online erreichbar und hat ihre eigenen Nutzer und *Akteure*.

Die Daten der einzelnen Stadtteilplattformen sind ausschließlich bei ihnen in persistierter Form vorhanden.

Aus Kapitel 2.3 wird geschlossen, dass die Installation und Konfiguration der einzelnen Plattformen möglichst einfach gehalten werden muss. Eine Trennung von Funktionalität und Aussehen der Website ist dabei unbedingt notwendig, genau wie eine externe Konfiguration der Plattform selbst. Abbildung 2 stellt solche isolierten Stadtteilplattformen dar. Man kann erkennen, dass sie keinerlei Abhängigkeiten voneinander haben.

3.2.3 Datenaustausch

Aus Kapitel 3.2.2 geht hervor, dass die Stadtteilplattformen keine Abhängigkeiten voneinander haben. Einzig die gemeinsame Codebasis verbindet sie miteinander.

Jeder dieser Instanzen beinhaltet Daten, die anderen Applikationen zur Verfügung gestellt werden sollen. Sie sind nach dem **Heloise Common Research Model** dem Repository Layer zuzuordnen.

Um einen Datenaustausch zu gewährleisten muss demnach eine weitere Komponente hinzugefügt werden. Diese Komponente ist dann dem Application Layer zugehörig und bietet den Zugriff auf die

³erreichbar auf <https://github.com/Juliane/easteast>

⁴<https://www.drupal.org/>

Daten der einzelnen lokalen Instanzen an.
Bei dieser Komponente handelt es sich um das semantische Portal.

3.2.4 Das Semantische Portal

In [MSS⁺02] werden die grundlegenden Aufgaben von Semantischen Portalen in drei Punkten beschrieben.

- Data Sources - Datenbeschaffung
- Integration - Mediatoren um die Daten zu kombinieren
- Wrapper - Datentransformation

Diese Beschreibung deckt sich ziemlich genau mit den im Application Layer des **Heloise Common Research Model** definierten Applikationen (siehe Kapitel 3.1).

Die Daten der einzelnen unabhängigen Datenbasen werden beschafft und in eine allgemeine Struktur transformiert. Somit können andere Applikationen auf die Daten zugreifen, ohne deren eigentlichen Implementierungen (im Repository Layer) kennen zu müssen. Gleichzeitig liegen die Daten durch die Datentransformation und Integration in einer bereits aufbereiteten Form vor.

Besonders bei der Datenintegration gibt es dabei verschiedene Ansätze.

- Generic Integration - Direkter Zugriff auf Daten
- Interconnected Integration - Weiterleitung zur eigentlichen Datenquelle
- Bounded Integration - Nutzung von Replikaten der eigentlichen Datenquelle

Darin wird unterschieden, wie die Daten bereitgestellt werden. Die Generic Integration setzt dabei die Möglichkeit des direkten Zugriffs auf die Daten und deren Struktur voraus, während die Bounded Integration eine Spiegelung der Daten vorsieht. Die Interconnected Integration sieht dabei keine Transformation der Daten vor und leitet Anfragen direkt an das eigentliche Repository weiter.

Die Datentransformation beschäftigt sich damit die Daten in möglichst vielen und verbreiteten Ontologien bereitzustellen. Aus diesem Grund werden *semantische Portale* auch als *Ontology Broker* bezeichnet.

Davies bezeichnet dabei den Vorteil von *semantischen Portalen* folgender Maßen:

”The benefit of an SW portal is that it is able to load this initial ontology and build a system out of the box that can satisfy user needs. It will be custom tailored but still be standard compliant.” [DSL⁺05, S.45]

Mit der Nutzung eines *semantischen Portales* wird ein zentraler Ansatz des Knowledge Managements gewählt [PB06]. Damit geht einher, dass das Portal eine sehr hohe Verfügbarkeit haben muss.

3.3 Konkreter Entwurf

3.3.1 Allgemeine Architektur

Abbildung 3 zeigt die Architektur der stadtteilübergreifenden semantischen Plattform, abgebildet auf dem Schichten Model des **Heloise Common Research Model**.

Das Schaubild zeigt drei Komponentenkategorien.

- Lokale Instanzen der Stadtteilplattform
- Das Semantische Portal
- Access Tools

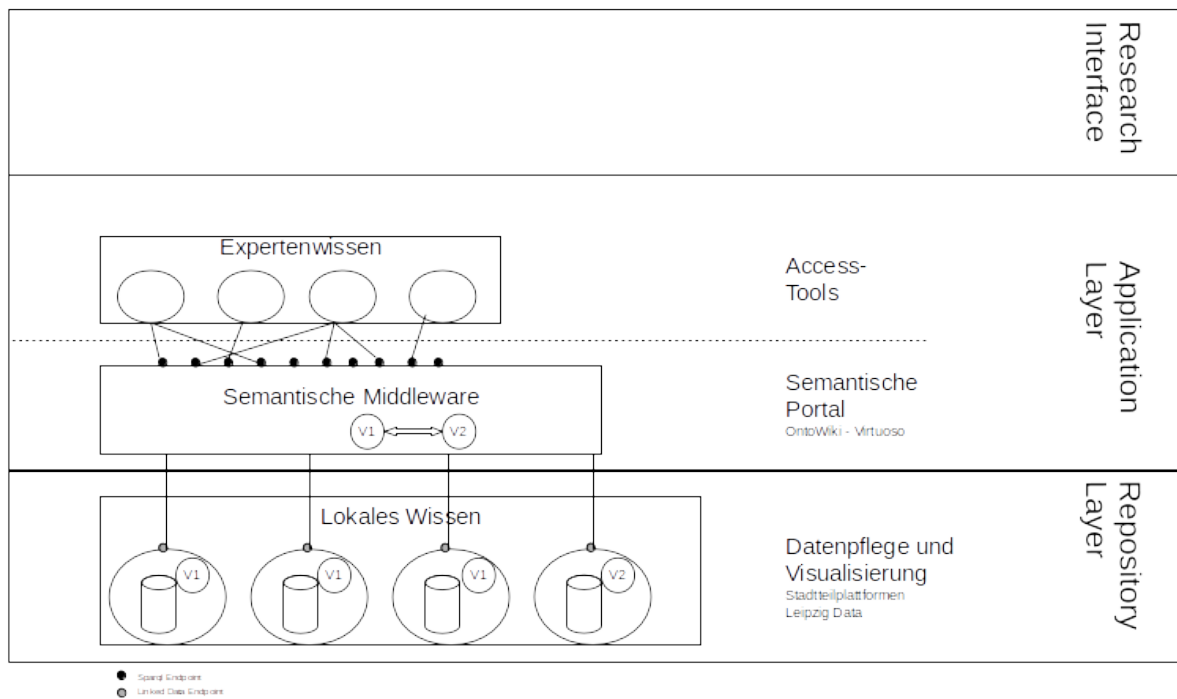


Figure 3: Schaubild der Softwarearchitektur des Semantischen Portals abgebildet auf dem **Heloise Common Research Model**

Es sei erwähnt, dass sich alle Komponentenkategorien ausschließlich auf Layer 1 und 2 des Schichtenmodells beziehen. Wie in Kapitel 3.1 beschrieben, handelt es sich demnach um die Repositories, also den Datenbasen, und Tools, die den Zugriff auf diese Daten ermöglichen.

Visualisierung und Datenpflege Im **Repository Layer** befinden sich die lokalen Instanzen der Stadtteilplattform. Diese sind, wie in Kapitel 3.2.2 beschrieben, vollkommen isoliert voneinander. Die lokalen Instanzen der Stadtteilplattform sind für die Visualisierung sowie der Pflege Daten der einzelnen Stadtteilplattform zuständig. Letztere wird vollständig von den Nutzern der jeweiligen Plattform selbstständig übernommen. Die Darstellung der Daten wird dann von den Plattformen selbst übernommen.

Darüber hinaus stellen die lokalen Instanzen ebenfalls die bereitgestellten Daten weiterer Stadteile dar. Die Art und Weise der Darstellung sowie die Auswahl der präsentierten Daten liegt dabei in der Verantwortung der Betreiber der Stadtteilplattform.

Jeder dieser Instanzen muss Zugriff auf ihre Daten gewähren. In der Grafik werden dementsprechend an jeder dieser lokalen Instanzen **Linked Data Endpoints** definiert. Diese stellen die Daten der jeweiligen Instanz in einer vortransformierten Form bereit.

Das bedeutet, dass die Daten bereits als Tripel in Form einer im Semantic Web verbreiteten Datenstruktur vorliegen. Als Beispiele seien hier RDF und Turtle genannt. Eine Beschreibung der Datenformate findet in dieser Arbeit nicht statt, kann jedoch unter [C⁺14] und [BBLP11] nachgelesen werden.

Das Semantische Portal als Middleware Im **Application Layer** befinden sich 2 weitere Komponenten. Einerseits ist das **semantische Portal**, andererseits einzelne **Access Tools** erkennbar.

Eine Beschreibung des **Semantischen Portals** kann in Kapitel 2.2 eingesehen werden. Exakt diese Funktionen müssen auch in diesem Anwendungsfall abgedeckt werden.

Genauer spezifiziert werden, kann dabei die Art der Integration. In dem Anwendungsfall der vorliegenden Arbeit wird von dem Konzept der Bounded Integration Gebrauch gemacht.

So stellt jede lokale Instanz im **Repository Layer** seine Daten, bereits vortransformiert als Turtle-Dateien, über einen Endpunkt bereit. Das **semantische Portal** konsumiert diese Endpunkte. Folgend werden die Daten weiterverarbeitet und aggregiert. Diese Weiterverarbeitung beinhaltet eine Kuration der Daten sowie das Abbilden der Daten auf weitere Vokabulare. Eine genauere Beschreibung dessen ist unter Kapitel 3.3.2 nachzulesen.

Darüber hinaus bietet das **semantische Portal** einen **SPARQL-Endpunkt** an. Dieser ermöglicht es, die bereits transformierten und kurierten Daten über die Abfragefrage SPARQL zu nutzen. Eine Beschreibung der Abfragesprache kann unter [HSP13] nachgelesen werden.

Die Rolle des **semantischen Portals** wird dabei von dem bereits existierenden Portal *Leipzig Data*⁵ übernommen. Es bietet bereits eine Vielzahl von Funktionen, wie einen **SPARQL-Endpunkt**, ein Ontowiki⁶ und verschiedene Vokabulare zur Darstellung semantischer Daten über Leipzig. Es wird nicht genauer auf *Leipzig Data* eingegangen.

Datenzugriff über Drupalmodule Die **Access Tools** nutzen den **SPARQL-Endpunkt** um die kurierten Daten wiederzugeben. Sie sind als eigenständige Tools einsetzbar. Die Weiterverarbeitung der vom **semantischen Portal** bereitgestellten Daten obliegt damit ihnen selbst.

Damit die lokalen Instanzen der Stadtteilplattformen Nutzen aus den kurierten und aggregierten Daten ziehen können, ist es wichtig, dass sie die **Access Tools** in diese eingebunden werden können. Somit

⁵Zu finden unter <https://www.leipzig-data.de>

⁶<http://aksw.org/Projects/OntoWiki.html>

werden sie in dieser Arbeit als **Drupal Module** definiert, da die lokalen Instanzen auf dem Drupal-CMS basieren. Erst dann können die einzelnen Stadtteilplattformen die Daten der weiteren Stadtteile darstellen.

Die semantische Stadtteilplattform Die Gesamtheit aller Komponentenkategorien bildet die semantische Stadtteilplattform. Sie ist wie in Kapitel 3.2.1 beschrieben stadtteilübergreifend. Darüber hinaus aggregiert sie die Daten der einzelnen Plattformen und transformiert sie so, dass sie in möglichst großem Umfang und flexibel wiederverwendbar sind.

3.3.2 Datentransformation

Die wichtigsten Eigenschaften für das Semantic Web sind laut [PB06, S. 375ff]:

- Standards
- Daten in einer allgemein anerkannten Struktur (bspw. RDF, TURTLE)
- Nutzung bestehender Ontologien

Um die kuriierten und aggregierten Daten der **semantischen Stadtteilplattform** möglichst flexibel zu nutzen, müssen sie dementsprechend transformiert werden. Diese Transformation findet auf 2 Ebenen statt.

Vortransformation Nach Erstellung oder Bearbeitung der Daten liegen diese vorerst ausschließlich als Datensätze einer relationalen Datenbank vor. Diese Datensätze gilt es in ein für **linked Data** übliches Format zu überführen.

Die Aufgabe ist manuell äußerst zeitintensiv und fehleranfällig, sodass eine Automatisierung von Nöten ist. Diese kann mit Hilfe der Mapping-Sprache **R2Rml** und geeigneten Transformationstools geschehen. Auf die Mappingsprache wird hier nicht genauer eingegangen. Sie kann unter [DSC13] nachgelesen werden. Als Transformationstool kann von dem Autor **r2rml-parser**⁷ empfohlen werden.

Das Transformationstool kann über ein Drupal-Modul in regelmäßigen Abständen aktiviert werden. Durch die gemeinsame Codebases (siehe 3.2.2) ist somit sicher gestellt, dass jede lokale Instanz der Stadtteilplattform diese Transformation vornimmt.

Die Nutzung der genannten Mapping-Sprache und des Tools ermöglicht eine Konfiguration der Bereitstellung der Daten, ohne den Quelltext der einzelnen lokalen Instanzen anzupassen. Somit kann jeder Stadtteil selbst entscheiden, welche Daten zur Verfügung gestellt werden. Durch die wenigen relevanten Entitäten der Stadtteilplattformen ist ein manuelles Schreiben der Mappings ohne Probleme möglich. Auf die genaue Nutzung und Beispiele kann aufgrund des Umfangs der Arbeit leider nicht weiter eingegangen werden.

Die Aufgabe der **Vortransformation** ist es, die semantisch relevanten Informationen der relationalen Datensätze in das TURTLE-Dateiformat zu transformieren. Dabei werden vor allem Vokabulare von *Leipzig Data*⁸, aber auch weitere wie **geonames**⁹ oder **foaf**¹⁰ beachtet.

Kuration und Weiterverarbeitung Das semantische Portal konsumiert in regelmäßigen Abständen die **linked Data** Endpunkte der lokalen Instanzen. Sobald die Daten vorhanden sind, werden sie

⁷Erreichbar unter <https://github.com/nkons/r2rml-parser>

⁸Siehe <http://www.leipzig-data.de/Data>

⁹Siehe <http://www.geonames.org/ontology#>

¹⁰Siehe <http://www.foaf-project.org/>

auf weitere Vokabulare angewandt (gemappt). Es sind als wichtige und verbreitete Vokabulare bspw. folgende zu nennen.

- foaf
- lode/event
- Dublin Core
- Schema.org [GY16]
- Liste unter [Ger12]

Ist dies geschehen, müssen die jeweiligen Ressourcen einen **Kurationsprozess** unterzogen werden. Dieser soll sicher stellen, dass die Daten "sauber" vorliegen. Fehlerhafte Datensätze können auf fehlerhaften Datenquellen sowie Fehlverhalten der Vortransformationsprozesse basieren. Die Kuration muss zum derzeitigen Standpunkt manuell stattfinden.

4 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein Architekturkonzept vorgestellt, welches die Daten von separaten Stadtteilplattformen semantisch aufbereiten und zur Verfügung stellen kann. Dabei wurden drei Komponentenkategorien definiert und auf dem **Heloise Common Research Model** abgebildet um eine möglichst hohe Allgemeingültigkeit zu gewährleisten. Diese Kategorien sind

- Visualisierungs und Datenpflege Tools - Lokale Instanzen der Stadtteilplattform
- Semantisches Portal - Aggregation, Mapping und Kuration der Daten
- Access Tools - Module zum Nutzen der aggregierten Daten

Die einzelnen Plattformen basierend aber zwar auf einer gemeinsamen Codebasis, sind jedoch im Gesamtentwurf von abstrahierter Natur. Es ist ausschließlich notwendig, dass sie einen **Linked Data** Endpunkt zur Verfügung stellen, über den sie Ihre Daten in vortransformierter Art bereit stellen. Diese Vortransformation ermöglicht es einen Zugriff auf die technische Implementation der einzelnen Plattform zu unterbinden und somit einen minimalen Zugriff zu benötigen. Das erhöht die Unabhängigkeit der einzelnen Plattformen.

Um die Daten vorzutransformieren wurden in dem Beispiel als Mappingsprache **r2rml** und als dazugehöriges Transformationstool **r2rml-parser** empfohlen. Bei dem verhältnismäßig geringen Ausmaß von semantisch relevanten Entitäten sind diese Tools vom Funktionsumfang ausreichend und von der Handhabung einfach genug.

Das Semantische Portal aggregiert die Daten der einzelnen Plattformen und bildet sie auf weitere verbreitete Vokabulare ab. Somit wird die Verantwortung der einzelnen Plattformbetreiber reduziert, was dem in Kapitel 2.3 beschriebenen Soziokulturellen Hintergrund zu Gute kommt.

Die genannte Verantwortung wird auf das **semantische Portal** ausgelagert. Hier muss das Mapping und die Kuration der Daten stattfinden. Mit Mapping ist hierbei die Abbildung der Daten auf weit verbreitete Vokabulare gemeint. Die Kuration bezieht sich auf die Kontrolle der Datensätze. Durch die automatisierte Vortransformation und fehlerhafte Datenquellen können fehlerhafte semantische Daten resultieren. Diese gilt es zu identifizieren und wenn möglich zu verbessern oder zu entfernen.

Um die Aufbereiteten Daten zu konsumieren, stellt das **semantische Portal** einen **SPARQL-Endpunkt** bereit. Dieser ermöglicht es weiteren Tools die aggregierten und aufbereiteten Datensätze flexibel zu nutzen. Diese Tools werden im Architekturentwurf als **Access Tools** bezeichnet. Sie müssen für ihren jeweiligen Zweck eigens entwickelt werden.

Um den oben genannten Verantwortlichkeiten gerecht zu werden muss weiterhin ein Betreiberkonzept

entstehen. Dieses legt fest, welche Personen oder Organisationen sich um welche Komponentenkategorie kümmert und deren Verantwortung inne hat. Da als **semantisches Portal** von bf Leipzig Data die Rede war, liegt nahe, dass deren Betreiber darin einen Platz finden werden. Eine genaue Ausarbeitung dessen und, wer eine lokale Instanz betreibt, wie diese dem **semantischen Portal** bekannt gemacht wird, und wer Access Tools erstellt gilt es in zukünftigen Arbeiten zu erstellen.

Weiterhin ist interessant, wie die Kuration der Daten automatisiert stattfinden kann. Eine Prüfung auf bspw. Richtigkeit oder Redundanzen der Resource-URI ist dabei vorstellbar. Weiterhin können Pflichtangabe eine Rolle spielen, um nur einige Ideen zu nennen.

Darüber hinaus ist eine Evaluation von anerkannten und verbreiteten Vokabularen sinnvoll, um das Mapping der Daten zu vervollständigen.

Allenfalls bietet die hier vorgestellte Architektur eine Menge von Erweiterungspunkten und Szenarien und kann auf verschiedene Domänen abgebildet werden. Durch die Abbildung auf das **Heloise Common Research Model** ist sie gut beschreibbar und kann verallgemeinert werden.

References

- [BBLP11] David Beckett, Tim Berners-Lee, and Eric Prud'hommeaux. Turtle-terse rdf triple language. *W3C Team Submission*, 14:7, 2011. Accessed: 29.02.2016.
- [C⁺14] World Wide Web Consortium et al. Rdf 1.1 concepts and abstract syntax. 2014. Accessed: 29.02.2016.
- [DSC13] Souripriya Das, Seema Sundara, and Richard Cyganiak. R2rml: Rdb to rdf mapping language, w3c recommendation 27 september 2012, 2013. Accessed: 28.02.2016.
- [DSL⁺05] John Davies, York Sure, Holger Lausen, Ying Ding, Michael Stollberg, Dieter Fensel, Rubén Lara Hernández, and Sung-Kook Han. Semantic web portals: state-of-the-art survey. *Journal of knowledge Management*, 9(5):40–49, 2005.
- [Ger12] LS3 Systems Institut AIFB Universität Karlsruhe D-76128 Karlsruhe Germany. Semantic Web - Ontologies. <http://semanticweb.org/wiki/Ontology>, 2012. [Online; accessed 26-January-2016].
- [GY16] Yahoo Inc. Microsoft Corporation Google, Inc. and Yandex. Schemas - schema.org. <http://schema.org/docs/schemas.html>, 2016. [Online; accessed 26-January-2016].
- [HSP13] Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. Sparql 1.1 query language. *W3C Recommendation*, 21, 2013. Accessed: 29.02.2016.
- [MSS⁺02] Alexander Maedche, Steffen Staab, Rudi Studer, York Sure, and Raphael Volz. Seal - tying up information integration and web site management by ontologies. *IEEE Data Eng. Bull.*, 25(1):10–17, 2002.
- [PB06] Tassilo Pellegrini and Andreas Blumauer. Semantic web. *Wege zur vernetzten Wissensgesellschaft. Berlin [ua] Springer*, 2006.
- [Rie] T. Riechert. Heloise schichtenmodell. Accessed: 28.02.2016.
- [War07] Daniel Warner. *Advanced SQL*. Franzis-Verlag GmbH, 2007.
- [wWe] Helden wider Willen e.V. Leipziger ecken. Accessed at 2016.02.28.