Ein neues Konzept für die semantische Suche in heterogenen Informationssystemen zu Fragestellungen aus Umwelt und Energie

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
der Fakultät Maschinenbau
Karlsruher Institut für Technologie (KIT)

genehmigte
Dissertation
von

Thorsten Schlachter

Tag der mündlichen Prüfung 09.10.2018

Hauptreferent: Prof. Dr.-Ing. Dr. h. c. mult. Georg Bretthauer
Korreferentin: Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova

Kurzfassung

Das Ziel der vorliegenden Arbeit besteht darin, ein neues Konzept für die semantische Suche in heterogenen Informationssystemen zu Fragestellungen aus Umwelt und Energie zu entwickeln, d.h. die Konzeption und Entwicklung einer Suchfunktion für Webportale, die zwar für den Nutzer so einfach wie herkömmliche Internet-Suchmaschinen funktioniert, jedoch qualitativ bessere, ggf. mehr Ergebnisse liefert als eine konventionelle Volltextsuche.

Dazu werden, ausgehend von einer Grundarchitektur, vier Architekturvarianten entworfen, vorgestellt und in konkreten Umsetzungsbeispielen evaluiert.

Schlagwörter: Semantik, Suche, Suchmaschine, Systemarchitektur, Umwelt, Energie

Abstract

The aim of this work is to develop a new concept for semantic search in heterogeneous information systems on the domains of the environmental and energy, that is, the conceptual design and development of a search function for web portals, which works as simple as conventional internet search engines, but provides qualitatively better, possibly more results than a conventional full-text search.
To this end, four architectural variants are designed out of a basic architecture and are evaluated in concrete implementation examples.

Keywords: semantics, search, search engine, system architecture, environment, energy

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

Ajax	Asynchronous JavaScript and XML
API	Application Programming Interface (Programmierschnittstelle)
APK	Android Package
APPX	Microsoft App Package
CMS	Content Management System
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DOC	Microsoft Word-Datei
DTD	Document Type Definition (Dokumenttypdefinition)
EnEG	Energieeinsparungsgesetz
EnEV	Energieeinsparverordnung
FADO	Fachdokumente Online
FTP	File Transfer Protocol
GEMET	General Multilingual Environmental Thesaurus
GSA	Google Search Appliance
HATEOAS	Hypermedia as the Engine of Application State
HTML	Hypertext Markup Language (Hypertext-Auszeichnungssprache)
HTML5	Hypertext Markup Language Version 5
HTTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IAI	Institut für Angewandte Informatik
ID	Identifier
IOSB	Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung
IoT	Internet of Things
IPA / .ipa	iOS application archive
ISS	International Space Station
JCP	Java Community Process
JSON	JavaScript Object Notation

JSR	Java Specification Request (im Rahmen eines Java Community Process)
JSR-286	Java Portlet Specification 2.0
KIT	Karlsruher Institut für Technologie
LUBW	Landesanstalt für Umwelt Baden-Württemberg, bis 2017 Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg
MAT	Mensch/Aufgabe/Technik
OGC	Open Geospatial Consortium
OWL	Ontology Web Language
PaaS	Platform as a Service
PDF	Portable Document Format
PM10	Particulate Matter 10µ
RDF	Resource Description Framework
RDFa	RDF in Attributes
RDF(S)	Resource Description Framework Schema (RDF Schema)
REST	Representational State Transfer
RSS	Really Simple Syndication
SaaS	Software as a Service
SASS	Syntactically Awesome Style Sheets
SKOS	Simple Knowledge Organization System
SNS	Semantic Network Service
SOA	Serviceorientierte Architektur
SOAP	Simple Object Access Protocol
SOS	Sensor Observation Service
SMTP	Simple Mail Transfer Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SUI	Semantische Suche nach Umweltinformationen
UDDI	Universal Description, Discovery and Integration
UM	Ministerium für Umwelt, Klima und Energiewirtschaft Baden-Württemberg
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

UUID	Universally Unique Identifier
W3C	World Wide Web Consortium
WMS	Web Map Service
WSDL	Web Service Definition Language
WWW	World Wide Web
XML	Extensible Markup Language

Vorwort

Das Internet bietet gewaltige Mengen von Informationen. Viele davon lassen sich über Internet-Suchmaschinen finden, andere sind für Suchmaschinen nicht erreichbar und stehen daher auch deren Nutzern nicht zur Verfügung. Die Gründe für die Nichterreichbarkeit von Daten sind vielfältig und lassen sich durch die Betreiber von großen Internet-Suchmaschinen nur teilweise, und wenn, dann oft mit hohem Aufwand, lösen. So vielfältig die Probleme beim Zugriff auf Daten sind, so vielfältig sind auch die Daten selbst. Sie unterscheiden sich beispielsweise im Grad ihrer Strukturierung, in Datentypen, Datenformaten, Schnittstellen, Systemen, Sprachen etc. – eine Vielfalt, die es für Suchmaschinenbetreiber schwer macht, mehr als kleine gemeinsame Nenner für alle Daten zu finden. Ein gemeinsamer Nenner kann zum Beispiel der Typ der Daten sein, z.B. Bilder (Bilddateien), die wiederum in einer Vielzahl von Datenformaten vorliegen, unterschiedliche Qualitäten von Metadaten beinhalten oder denen bestimmte Lizenzen/Urheberrechte zugeordnet sein können. Internetsuchmaschinen wie Google bieten dann z.B. basierend auf dem Datentyp „Bild“ eine entsprechende Bildersuche an und versuchen die Präsentation der Suchergebnisse möglichst unabhängig, aber soweit wie möglich unter Einbeziehung des Datenformats, der Bildgröße, der Metadaten oder der Lizenz, zu gestalten. Dabei sind meist Kompromisse notwendig und Suchmaschinenbetreiber stecken erheblichen Aufwand in die automatisierte Klassifikation von Daten, z.B. dem Erkennen, was auf einem Bild dargestellt ist – der Semantik (Bedeutung) der Daten. Wer die Bedeutung der Daten versteht, kann auch eine Suche nach ihnen leichter umsetzen.

In abgeschlossenen Bereichen, z.B. den Datenbeständen einer Behörde, ist die Vielfalt der Daten geringer als im gesamten Internet, allerdings werden auch dort eine Vielzahl von Datentypen, Systemen, Datenformaten etc. verwendet. Ein Vorteil von abgeschlossenen Bereichen gegenüber dem Internet ist jedoch häufig, dass die Daten sich auch inhaltlich auf einen beschränkten Bereich (eine inhaltliche Domäne) beziehen. Das heißt, dass bei der Klassifizierung von Daten lediglich innerhalb der Domäne zugeordnet werden muss. In vielen Fällen wird sogar ein einheitliches Wortgut (Vokabular) zur Beschreibung der Daten verwendet, manchmal sind sogar die Schemas der Datensätze oder gemeinsame Schlüssellisten (einheitlich) festgelegt. Das erleichtert das eindeutige Zuordnen von Daten zu Themen und bietet damit ein großes Potenzial für das präzise Auffinden von Daten zu einem bestimmten Thema oder die Darstellung von mit den Daten im Zusammenhang stehenden weiteren Daten.

In der vorliegenden Arbeit geht es um das Auffinden von Daten innerhalb abgegrenzter Domänen, insbesondere aus den Bereichen „Umwelt“ und „Energie“. Hier soll für einen begrenzten, dennoch großen Datenbestand das Potenzial einer semantischen Suche geprüft und in verschiedenen Umsetzungsvarianten realisiert werden. Dazu sollen Daten aus dem Bereich der Umweltverwaltung herangezogen werden, auch da er in sei-

ner Vielfalt von Datentypen und Formaten nahezu den Herausforderungen von Internetsuchmaschinen entspricht, dennoch eine relativ abgegrenzte Domäne bildet. Bestandteil der Domäne „Umwelt“ (Koch und Frees 2015)ist teilweise auch der thematische Bereich „Energie“, der Schnittmengen mit der Umwelt hat, jedoch auch als eigenständige Domäne gelten kann.

Anspruch der vorliegenden Arbeit ist es, die entwickelten informationstechnischen Architekturen und Verfahren auch in der Praxis zu erproben, d.h. in konkreten Projekten umzusetzen. Das bedeutet, dass sie nicht nur grundsätzlich oder im Ansatz funktionieren, sondern auch einen Praxistest bestehen müssen.

Die vorliegende Arbeit wäre nicht möglich gewesen ohne die Unterstützung, Aufmunterung, Geduld und Hartnäckigkeit verschiedener Personen, insbesondere meines Betreuers Prof. Dr.-Ing. Dr. h. c. mult. Georg Bretthauer, meiner Chefs und Kollegen Dr. Clemens Düpmeier, Dr. Werner Geiger, Rainer Weidemann, Eric Braun, Claudia Greceanu, Christina Grieß, Christian Schmitt, Gerd Zilly und Prof. Dr. Veit Hagenmeyer.

In vielen Projekten, die in die vorligende Arbeit eingeflossen sind, durfte ich mit Kollegen anderer Institutionen an spannenden Themen zusammenarbeiten. Stellvertretend für viele weitere danke ich Wolfgang Schillinger (LUBW), Martina Tauber (LUBW), Renate Ebel (LUBW, IM BW), Roland Mayer-Föll (UM BW), Kurt Weissenbach (UM BW), Fernando Chaves Salamanca (IOSB), Ulrich Bügel (IOSB), Andreas Abecker (FZI, disy), Thomas Sattler (DECON-network), Joachim Fock (Umweltbundesamt), Thomas Bandholtz (ehem. innoQ), Lars Koch (CONVOTIS AG) sowie den vielen Partnern aus der LUPO-Kooperation.
Meiner Frau Stephanie und meinem Sohn David bin ich zu Dank verpflichtet, da sie phasenweise wenig von ihrem Mann bzw. Papa hatten.
Zuletzt möchte ich auch meinen Eltern danken, die mir in vielen Phasen meines Studiums und meines Lebens den Rücken freigehalten haben.

1 Einleitung

1.1 Bedeutung der semantischen Suche in heterogenen Informationssystemen

Aus Sicht eines Nutzers steht und fällt der Nutzen von Webportalen (s. Anhang A3) und Informationssystemen mit der Frage, wie schnell und mit welchem Aufwand er oder sie an die gewünschte Information gelangt bzw. die gewünschte Aufgabe erledigen kann. Die Art und Weise der Internetnutzung und Online-Recherche hängt dabei stark von der betrachteten Nutzergruppe ab. Nach der Online-Studie von ARD und ZDF (Koch und Frees 2015) waren zumindest knapp 80% der deutschen Bevölkerung im Jahr 2015 zumindest gelegentlich online, der Anteil der täglichen Internet-Nutzer lag bei rund 63%[1]. Während 76% der deutschen Internet-Nutzer angeben, das Internet zur Suche nach Informationen zu verwenden, verwenden sogar 82% Online-Suchmaschinen, um an die gewünschten Daten zu gelangen, was eindrucksvoll den Erfolg und Nutzen von Internet-Suchmaschinen zeigt. Nicht nur aus der Sicht der Nutzer stellen Suchmaschinen zentrale Einstiegspunkte ins Web dar, sondern auch Informationsanbieter müssen sich um ein hohes Ranking und eine entsprechend gute Platzierung auf den Trefferseiten der Suchmaschinen bemühen. Nach (Lunapark 2015) lag der weltweite Marktanteil der Suchmaschine *google.com* im Jahr 2015 bei über 90%, in Deutschland sogar bei über 93%.

Offenbar setzt Googles Suchmaschine also einige wesentliche Erfolgsfaktoren um, die ihre Nutzung für eine große Zahl von Internet-Nutzern attraktiv macht.

Nach (Wandiger 2009) lässt sich der Erfolg von Google kurz als „einfache und umfassende Suche“ beschreiben. Erfolgsfaktoren seien dabei

- die simple Nutzung und Bedienoberfläche („single search slot“, mobile enabled user interface, ...),
- die hohe Relevanz der Suchergebnisse,
- das Erkennen, was gesucht wird (Semantik der Anfrage),
- die Nutzung relevanter, hochwertiger Datenquellen,
- Kompetenz und Glaubwürdigkeit der Suchmaschine,
- gute Ranking-Faktoren,
- die Integration von Einzeldiensten (Volltextsuche, Nachrichten, Medien, Karten, Zeitreihen, ...),

---

[1] Dabei spielt die Nutzung mobiler Endgeräte eine zunehmend größere Rolle. 23% der Bevölkerung nutzten das Internet im Jahr 2015 täglich von einem Mobilgerät aus, 55% zumindest gelegentlich. Mobile Endgeräte werden dabei vornehmlich durch jüngere Personen im Alter unter 50 Jahren genutzt.

- Data Mining (das Erschließen großer Datenbestände; inklusive dem Sammeln von Wissen über den Informationsbedarf des einzelnen Nutzers),
- Offenheit (Openness, das Teilen von Inhalten),
- der Zugang zu verwandten Themen und
- die (für den Nutzer) kostenfreie Nutzung.

Allerdings hat die Recherche mit Internet-Suchmaschinen auch Grenzen. So werden große Teile des Web und andere, grundsätzlich online verfügbare Datenquellen nicht durch Suchmaschinen indexiert und sind daher nicht über die klassischen Suchmaschinen erreichbar. Man spricht vom „Invisible Web“, „Hidden Web“, „Darknet“ oder „Deep Web“ - im Gegensatz zum sichtbaren „Surface Web“ (Sherman und Price 2001). Gründe für die Nicht-Indexierbarkeit gibt es viele, z.B. kontextabhängige Inhalte, dynamische Inhalte, limitierten Zugriff (Passwortschutz, Ausschluss von Suchmaschinen), spezifische/proprietäre Datenformate oder Softwareprodukte, Nutzung von Scripten, fehlende Links etc.
Untersuchungen von (Bergman 2001) schätzten ab, dass das Deep Web mehr als zwei Größenordnungen (Faktor 450-550) größer sei als der sichtbare Teil. Aktuellere Studien und Zahlen können die tatsächlich theoretisch verfügbare Datenmenge ebenfalls nur schätzen, gehen aber nach wie vor davon aus, dass der für Suchmaschinen erreichbare Teil des Web, Branchenprimus Google indexierte nach (WorldWideWebSize.com 2017) im Januar 2017 über 46 Milliarden Seiten/Dokumente, höchstens die Spitze eines Eisbergs darstellt. (Lewandowski und Mayr 2006; Lewandowski 2015; Wrigth 2009).

Auch wenn für einige der aufgeführten Probleme, z.B. die Ausführung von im Inhalt enthaltenem Programmcode, z.B. JavaScript, während der Indexierung, durchaus Lösungsansätze oder sogar Lösungen existieren, so gibt es für den größten Teil des Deep Web keine Zugangslösungen und die enthalten Inhalte werden vermutlich schon aufgrund ihrer Masse auch in absehbarer Zeit nicht von Internet-Suchmaschinen erreicht werden - durch die automatisierte Erzeugung von Daten („Internet of Things“ (IoT), Sensornetze) werden die Anzahl der Datenquellen und die Menge der erzeugten Daten sogar noch rasanter wachsen als bisher (Siemens 2014).

Die beschriebene Problematik für große Internet-Suchmaschinen bietet jedoch Chancen und Nischen für kleine, spezialisierte Suchmaschinen, die nicht den Anspruch haben, eine Recherche über das gesamte Web anzubieten, sondern sich auf einen spezifischen, thematisch (eng) eingegrenzten Bereich („Domäne“) spezialisieren. Für sie kann es sich lohnen, Aufwand in die Implementierung spezifischer Schnittstellen zu relevanten Informationen zu stecken, um organisatorisch/rechtlich Zugang zu bestimmten Daten zu bekommen oder um Daten aus verschiedenen Quellen miteinander zu verknüpfen.

In einer abgegrenzten, bekannten Domäne ist es zusätzlich erheblich leichter, auch die Semantik von Daten und damit auch Zusammenhänge zu erfassen (Gliozzo und Strapparava 2009). Während die meisten klassischen Internet-Suchmaschinen zu großen Teilen auf dem Vergleich von Zeichenketten basieren (Lewandowski 2015), kann eine

spezialisierte Suchmaschine zusätzlich sowohl die Semantik der indexierten Inhalte als auch die Semantik von Suchanfragen erfassen. Hilfreich kann hier auch die Erkennung bestimmter Muster in den Suchanfragen sein, z.B. die Verknüpfung eines thematischen Suchbegriffes mit einem Orts- oder Zeitbezug. So kann zum Beispiel das Erkennen des Schlagwortes “Bahn” im Zusammenhang mit zwei erkannten Städtenamen und einer Zeitangabe zur Suche nach Bahnverbindungen zwischen den beiden Orten genutzt werden - sofern die Suchmaschine über die entsprechenden Daten oder eine Schnittstelle zu einem entsprechenden Hintergrundsystem (Bahn-Fahrplan) verfügt.

Mit Hilfe einer erkannten Semantik und der daraus angeleiteten Zuordnung zu einem bekannten Muster ist eine qualitativ bessere Suche als mit klassischen, universellen Suchmaschinen möglich. Wesentliche Herausforderungen liegen dabei allerdings in der Repräsentation der Daten – die größten Teile des Web sind nach wie vor auf den Menschen als Endnutzer ausgerichtet (Hitzler et al. 2008), in der Abbildung von Anfragen und Inhalten auf bekannte Konzepte (Erfassung der Semantik) und Anwendungsfälle, in semantischen Inkonsistenzen innerhalb der Daten (über verschiedene Datenquellen hinweg) sowie in ihrer Darstellung (z.B. Nutzung verschiedener Formate, Attribute oder Einheiten). Erschwert wird die semantische Verarbeitung dadurch, dass viele Datenquellen keine explizite Semantik liefern und aus Sicht der Suchmaschine kein Einfluss auf die Datenquellen besteht.

Für die Nutzer aus der Zielgruppe „Öffentlichkeit“ muss die Suche so einfach und transparent nutzbar sein, wie sie es von klassischen Volltextsuchmaschinen gewohnt sind, und komplexe Technik innerhalb der Suchmaschine darf sich nicht in einer komplexen Nutzerschnittstelle wiederspiegeln.

Die vorliegende Arbeit richtet sich inhaltlich an den im Titel enthaltenen Schlagworten „Umwelt“ und „Energie“ aus, d.h. an zwei großen Themenbereichen, welche die oben angesprochenen Domänen bilden und eingrenzen. Selbstverständlich bieten beide Bereiche spezifische Probleme. Viele Fragestellungen und Probleme sind jedoch grundsätzlicher Art, d.h. man findet sie domänenübergreifend wieder, oder es bestehen Beziehungen zu Bereichen außerhalb der Domäne, z.B. zwischen der Domäne „Umwelt“ und den Domänen „Energie“, „Wirtschaft“, „Politik“ etc.
Die beiden Domänen reißen bereits ein breites Spektrum möglicher Anwendungsszenarien auf. Jedes Szenario beinhaltet dabei Zielgruppen (z.B. Öffentlichkeit, (politische) Entscheider, Fachleute oder Techniker), Fragestellungen und Erwartungen.

Die entwickelten Konzepte und Technologien wurden in mehreren Kontexten erprobt, z.B. in den Domänen „Umwelt“ und „Energie“, es flossen jedoch auch Erfahrungen aus Anwendungen in anderen Kontexten und Domänen, z.B. aus den Bereichen der Jugendbildung und Jugendarbeit, in die Arbeit ein. Der überwiegende Anteil der Beispiele kommt jedoch aus den Domänen „Umwelt“ und „Energie“, da die Grundtypen ähnlich sind und viele Informationen aus dem Umweltbereich sich durchaus auch mit der Domäne “Energie” verknüpfen lassen, für viele Anwendungsszenarien drängen sich solche Verknüpfungen sogar auf - nicht erst seit der Atomkatastrophe in Fukushima, sondern bereits seit über drei Jahrzehnten, dem Beginn des Übergangs der Energiever-

sorgung hin zu mehr Nachhaltigkeit (Krause et al. 1981), insbesondere durch die vermehrte Nutzung von erneuerbaren Energieträgern in den Sektoren Strom, Wärme und Verkehr/Mobilität. Die „Energiewende“, zunächst nur ein Schlagwort, das im Folgenden als Beispiel für den größeren Bereich der Energie dienen soll, betrifft dabei sehr viele Lebensbereiche und muss daher als gesamtgesellschaftliche Aufgabe verstanden werden. Für die verschiedensten Gruppen (Energiewirtschaft, Wirtschaft, politische Entscheidungsträger, Forschung und Entwicklung, interessierte Bürger etc.) entstehen neue Fragestellungen, die wesentliche Auswirkungen auf deren Zukunft und das tägliche Leben haben werden. Die Bandbreite der Fragen reicht dabei von der Versorgungssicherheit über künftige Energiepreise, Geschäftsmodelle, Energieeinsparung, technologische Aspekte bis hin zu Wechselwirkungen mit anderen Bereichen wie z.B. Umwelt, Bauen, Lebensmittelversorgung, Verkehr, Arbeit etc.
Viele Ziele der Energiewende haben einen direkten Bezug zur Umwelt und viele Auswirkungen der Energiewende werden sich daher auch im Umweltbereich zeigen bzw. Kriterien für die Zielerreichung lassen sich durch die Beobachtung der Umwelt bestimmen und/oder messen, z.B. die Begrenzung der globalen Temperatur oder die Verbesserung der Luftqualität. Daneben stehen viele Entscheidungen im Bereich der erneuerbaren Energien in einem direkten oder indirekten Zusammenhang mit Umweltfragen, z.B. beim Flächenverbrauch, Bauen in Schutzgebieten, Wasserverbrauch für Kühlwasser, Stauung von Fließgewässern, Nutzung von Biomasse zur Energieerzeugung (statt als Dünger) etc. Umwelt und Energie stellen also zwei eng inhaltlich miteinander verwobene Domänen dar, die auch jede für sich betrachtet werden kann.

Die Verknüpfung bzw. die Verknüpfbarkeit von Energie- und Umweltinformationen ist für die Klärung vieler Fragen zur Wirksamkeit des Einsatzes erneuerbarer Energien bzw. zur Energiewende von essentieller Bedeutung.

Das „Umweltinformationssystem Baden-Württemberg“ (UIS BW) bietet mit einer Vielzahl von Systemen und Datenbeständen bereits grundsätzlich Zugang zu vielen behördlichen Umweltinformationen und -daten, z.B. Labor- und Stoffdaten, statistischen Daten, Geobasisdaten, Geofachdaten, Fachdokumenten, Messdaten, Metadaten etc. (Mayer-Föll 1992, 1993).
Es fehlt dem UIS BW jedoch bislang eine Suchfunktion, die Daten aus der Vielzahl von Einzelsystemen des UIS BW zusammentragen und in übergreifender Art und Weise darstellen kann.

Die zentrale Forschungsfrage beschäftigt sich daher mit der Verbesserung der Suchergebnisse innerhalb der Domäne „Umwelt“ und „Energie“:

> *„Wie kann ein Nutzer eines Webportals in einer heterogenen Landschaft von existierenden Web- und Informationssystemen durch simple Suchanfragen („single search slot-Philosophie“) zu Fragestellungen aus den Bereichen „Umwelt“ und „Energie“ umfassende (semantisch stimmige) integrierte Antworten erhalten?“*

Der Begriff „Nutzer“ wird hier implizit durch den Zusammenhang mit der Formulierung von „simplen Suchanfragen“ eingeschränkt. Während Fachnutzern als Experten auf ihrem Gebiet eine komplexe, fachlich differenzierte Rechercheschnittstelle zugemutet werden kann, benötigen Laien, z.B. Bürger oder die „interessierte Öffentlichkeit“ Rechercheschnittstellen von möglichst geringer Komplexität - vom Erfolgsfaktor „single search slot“ von Google war ja bereits weiter oben die Rede. Ein Schwerpunkt bei der Auswahl von Anwendungsszenarien (Abschnitt 1.3.6) wird also im Bereich der „interessierten Öffentlichkeit“ liegen.

Unter der Formulierung „von existierenden Web- und Informationssystemen“ werden Systeme verstanden, die „so sind wie sie sind“, d.h. der Betreiber des Webportals inklusive der Suchmaschine hat keinen Einfluss auf die Inhalte und Form der von den Systemen bereitgestellten Daten und ggf. Metadaten. Die Daten werden in einer „heterogenen Landschaft“, d.h. über unterschiedliche Protokolle, Repräsentationen, technische Formate und Codierungen sowie verschiedene (implizite oder explizite) Semantiken verfügbar gemacht. Sie werden durch die Suchmaschine erfasst und im Portal in stimmigen Trefferansichten dargestellt, d.h. jedem Datentyp werden eine oder mehrere passende Darstellungsformen (Liste, Tabelle, Diagramm, Karte etc.) zugeordnet.

Das Gesamtziel der vorliegenden Arbeit ist also die Konzeption und Entwicklung einer Architektur für die Suchfunktion von Webportalen, die zwar für den Nutzer so einfach wie herkömmliche Internet-Suchmaschinen funktioniert, jedoch bessere, ggf. mehr Ergebnisse liefert als eine konventionelle Volltextsuche. Dabei sollten folgende wissenschaftlichen Teilziele erreicht werden (Abbildung 1):

- Erkennen der Semantik einer Suchanfrage innerhalb einer gegebenen Domäne
- Beschreibung der Semantik von Daten gegebener Informationssysteme bezüglich einer vorgegeben Domäne, die sich ggf. aus mehreren Vokabularen (s. Anhang A1.2) zusammensetzt:
    - Mapping der Vokabulare untereinander (Artikulation) bzw. Harmonisierung
    - Abbildung der Daten auf Vokabulare und ggf. Nutzung zugehöriger Schemata
    - Nutzung von Zusatzwissen zu Orts- und Zeitbezug, auch als generische Zusammenhänge zwischen Daten
    - Harmonisierung der Darstellung von Daten aus verschiedenen Informationssystemen
    - Nutzung der in den Vokabularen enthaltenen Semantik, z.B. zur Weiternavigation, Gruppierung, Facettierung etc.
- Beschreibung und Realisierung des technischen Zugriffs auf heterogene Informationssysteme (Datentypen, Schnittstellen und Formate)
- Nutzung generischer Komponenten zur Präsentation von Daten innerhalb eines Webportals
- Präsentation/Darstellung der Suchergebnisse in einer integrierten Trefferansicht
    - Unterschiedliche Darstellungen (Karte, Diagramm, Tabelle, Liste etc.)

- Möglichkeit zur Kommunikation unter den Darstellungskomponenten.

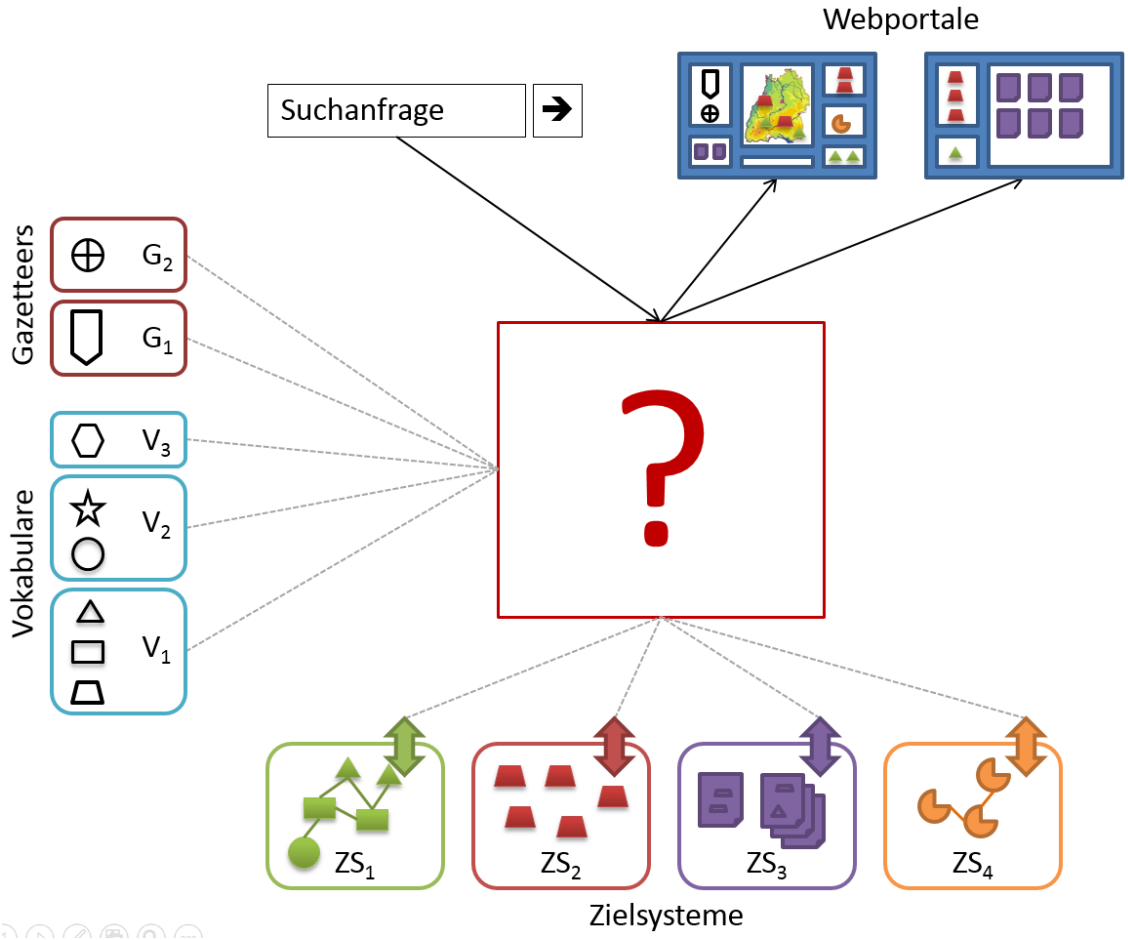

Abbildung 1: Komponenten einer Architektur für eine semantische Suche

Abbildung 1 zeigt den Rahmen einer allgemeinen Architektur für die semantische Suche und damit bereits eine Art Grundarchitektur: Die Zielsysteme ZS_i (unten) enthalten die Daten, jeweils in einer spezifischen Darstellung und Semantik. Die Zielsysteme enthalten im Allgemeinen verschiedene Datentypen, bieten unterschiedliche Schnittstellen zum Zugriff auf die Daten und verwenden dabei eine ganze Reihe technischer Formate („Anbindung Zielsystem“).

Dem Nutzer soll über ein oder mehrere Webportale (rechts oben) eine Recherchemöglichkeit für die in den Zielsystemen enthaltenen Daten geboten werden. Die minimalen Bestandteile der Benutzerschnittstelle sind ein Eingabefeld („Suchschlitz“) für das Formulieren der Suchanfrage (oben links) sowie die Präsentation der Suchergebnisse („Ergebnisdarstellung“), z.B. innerhalb von Webportalen, ggf. in einer integrierten Ansicht, welche adäquate Darstellungen (Karte, Diagramm, Liste etc.) für die verschiedenen Datentypen bietet.

Der zwischen Zielsystemen und Webportalen befindliche rote Kasten („?“) repräsentiert die im Rahmen der vorliegenden Arbeit zu lösenden Aufgabenstellungen, zunächst den technischen Zugriff auf die Zielsysteme und anschließend die semantische Verknüpfung zwischen den Suchanfragen der Nutzer und den in den Zielsystemen enthaltenen Daten. Mehrere Vokabulare V_i (links unten) können zur semantischen Harmonisierung

der Suchanfragen und Antworten verwendet werden. Sie dienen der Suche als Hintergrundwissen. Die einzelnen Vokabulare können dem semantischen Modell eines oder mehrerer Zielsysteme entsprechen, es sind jedoch auch zusätzliche Vokabulare möglich, z.B. zur Anknüpfung an weitere Domänen, übergreifende Vokabulare, Artikulationen (Verknüpfungen zwischen Vokabularen), Spezialisierungen etc.
Die Gazetteer-Dienste G_i (links oben) können Zusatzwissen zum Explizieren der in den Zielsystemen enthaltenen Daten bzw. der Suchanfragen vermitteln, z.B. um vorhandene Attribute wie ein Adressfeld in eine andere Repräsentation (Geokoordinaten) zu überführen oder eine Zeitangabe wie „Sommer 2012“ in explizite Datumsangaben (Beginn: 21.06.2012; Ende: 22.09.2012; Gebiet: Nordhalbkugel)[2] umzuwandeln.

Die Kriterien für das Erreichen von Zielen sind deshalb unter Berücksichtigung der verschiedenen Rollenverständnisse zu definieren (Abschnitt 1.3). Die Evaluation soll anhand ausgewählter Anwendungsszenarien erfolgen (Abschnitt 1.3.6).

Der Spielraum möglicher Architekturvarianten wird durch die gegebenen Voraussetzungen (Abschnitt 1.2) und existierende Nebenbedingungen (Abschnitt 1.3.6) eingeschränkt.

Die oben aufgeführten Ziele stehen jedoch nicht alleine, sondern müssen jeweils im Kontext verschiedener Rollen betrachtet werden:

- Nutzer von Webportalen (bzw. übergreifenden Informationssystemen)
- Betreiber solcher Portalsysteme
- Betreiber von Informationssystemen als originäre Datenquellen (Zielsysteme)
- Weitere (potenzielle) Nutzer der Daten / Übergreifende Interessen.

Für die Nutzer eines Webportals steht zunächst im Vordergrund, die gewünschte Information möglichst einfach finden zu können. Die Benutzeroberfläche sollte daher einfach und intuitiv bedienbar sein und mit verschiedenen Geräteklassen (PC, Laptop, Tablet, Smartphone) funktionieren. Je nach Verfügbarkeit sollen vorhandene Kontextinformationen, insbesondere der Standort des Nutzers, verwendet werden. Die Antwortzeit soll möglichst kurz sein und die Darstellung der Treffer soll übersichtlich und passend zum jeweiligen Typ der gefundenen Daten gestaltet sein.

Für die Betreiber von Portalen soll der Anschluss von Informationssystemen an das Portal möglichst einfach sein, d.h. in der Regel nur ein vertretbarer Konfigurationsaufwand - in Abgrenzung zum aufwändigen Programmieren von Schnittstellen - entstehen. Andererseits soll es möglich sein, existierende Informationssysteme in der Regel ohne Änderungen an den Systemen selbst zu integrieren, denn in den meisten Fällen wird der Betreiber des Portals keinen Einfluss auf ein zu integrierendes System haben, z.B. die Entwicklung einer spezifischen Schnittstelle zu erwirken. Zur Nutzung extern

---

[2] Der Schluss auf die Nordhalbkugel beinhaltet eine Voreinstellung (Default-Wert), welche der typischen Verwendung der Suche (in deutschsprachigen) Webportalen entspricht. Ggf. könnte sich die Interpretation durch weitere Informationen, z.B. eine in der Suchanfrage enthaltene Ortsangabe oder weitere Kontextinformationen (Standort des Nutzers), ändern.

verfügbarer Informationen wird in der Regel auch ein Mapping der externen Daten auf ein im Portal verwendetes Datenmodell notwendig sein, das ebenfalls per Konfiguration möglich sein soll.

Für die Betreiber von Informationssystemen als Quellsysteme für ein übergreifendes Portal ist es natürlich wichtig, dass zunächst die rechtlichen Rahmenbedingungen (Urheberrecht, Bildrechte, Nutzungsrechte, …) zur Nutzung der Daten innerhalb eines externen Portals geklärt sind. Im Idealfall stellen Systeme ihre Daten bereits über wohldefinierte Schnittstellen und standardisierte Datenformate zur externen Nutzung zur Verfügung. In allen anderen Fällen soll dem Betreiber eines Informationssystems in der Regel kein oder zumindest kein erheblicher Aufwand entstehen. In vielen Fällen bedeutet das, dass ein Quellsystem so benutzt werden muss wie es ist, z.B. über seine für menschliche Nutzer ausgelegte Web-Oberfläche. Das Datenmodell des Quellsystems sollte nicht geändert werden müssen. Für ein Mapping bzw. eine evtl. gewünschte semantische Erweiterung des Datenmodells muss es daher Lösungen auf Seiten des Portals geben.

Wenn Daten/Informationen aus verschiedenen Informationssystemen in einheitlichen Darstellungen innerhalb eines Portals verfügbar gemacht werden sollen, liegt die Frage nahe, ob es nicht möglich ist, die Daten allgemein über standardisierte Schnittstellen auch außerhalb des Portals bereitstellen zu können, um sie auch für andere Anwendungen nutzbar zu machen. Die am weitesten gehenden Überlegungen hierzu sind mit Begriffen wie „Open Data“[3] (Herb 2012), „Linked Data“ (Berners-Lee 2006) (s. Anhang A1.1) bzw. „Semantic Web“ (Berners-Lee und Fischetti 1999) (s. Anhang A1) verbunden. Das geht in seiner Konsequenz deutlich über die Ansprüche und Anforderungen der vorliegenden Arbeit hinaus. Die Überlegungen bieten jedoch Visionen einer interoperablen Welt von Informationen und Informationssystemen an, deren grundlegende Ideen und Ansprüche auch für die hier angestellten Überlegungen eine (langfristige) Richtschnur sein können.

1.2 Darstellung des Entwicklungsstandes von heterogenen Informationssystemen und der semantischen Suche

Mit der allgemeinen Verfügbarkeit des Internets, spätestens jedoch seit dem Aufkommen serviceorientierter Architekturen wurde die Integration heterogener Landschaften von Informationssystemen nicht nur zu einer praktischen, sondern ebenfalls zu einer wissenschaftlichen Herausforderung (Buxmann 1996). Neben den zuvor meist betrachteten technischen und syntaktischen Problemen, z.B. zwischen Datenbanksystemen (Conrad 1997) traten nun vor allem die heterogenen Semantiken und deren Verknüpfung in den Vordergrund (Wache 2003; Naiman und Ouksel 1995; Domingue et al. 2011).

---

[3] Hier soll nicht auf die damit verbundenen rechtlichen Fragestellungen eingegangen werden.

Bei den zu betrachtenden Zielsystemen (Abbildung 1) handelt es sich um eine Menge heterogener Informationssysteme, die sich in ganz unterschiedlichen Aspekten unterscheiden können, z.B.

- Datentypen (Art von Daten, Strukturierung, Semantik...)
- Schnittstellen (für Maschinen/Menschen)
- Datenformate/Repräsentationen (z.B. HTML, PDF, JSON, XML, WMS…)
- Semantik (definiert je Zielsystem, aber nicht übergreifend/global)
- Erreichbarkeit (z.B. fehlende Verlinkung, Dark Web)
- Verfügbarkeit (z.B. Vorhandensein von Wartungsfenstern)
- Skalierbarkeit (z.B. beschränkte Anzahl paralleler Nutzeranfragen).

Neben der semantischen Verknüpfbarkeit von Daten ist eine weitere wesentliche Herausforderung der semantischen Suche, dem Nutzer eine möglichst einfache und intuitive Benutzerschnittstelle zu bieten, gerade weil viele existierende Ansätze keine besonders einfachen – wenn dafür auch mächtige – Benutzerschnittstellen bieten, und sich damit eher an Experten als an Laien bzw. die interessierte Öffentlichkeit wenden (Mangold 2007; Bizer et al. 2009).

Obwohl seit Jahren Diskussionen über das Design von Benutzerschnittstellen für Suchfunktionen geführt werden (McKay 2011), muss man jedoch konstatieren, dass der Erfolg der Google-Suche sowie neuerer, natürlichsprachlicher Benutzerschnittstellen wie Siri (Apple Inc. 2017) oder Alexa (Amazon 2017) neben der Qualität der Suchergebnisse sicherlich auch auf der einfachen Bedienbarkeit und Intuitivität ihrer Benutzerschnittstellen beruht.

Die Suchanfragen an eine semantische Suche sollen daher einfach, z.B. in Form eines „single search slot“ formuliert werden können, d.h. sie werden in Form einer einzigen Zeichenkette an die Suchfunktion übermittelt; natürlichsprachliche Eingaben lassen sich mithilfe von Spracherkennung auf solche Zeichenketten abbilden. Dabei sollte es um eine rein inhaltliche Formulierung der Suche gehen, d.h. möglichst nahe an der natürlichen Sprache und ohne formale Syntax, wie z.B. SPARQL (W3C 2008).
Solche einfachen Benutzerschnittstellen und Suchanfragen stehen im Gegensatz zu komplexeren Suchformularen, die mehrere Suchschlitze/Formularfelder anbieten, in denen der Nutzer semantisch vorklassifizierte Suchbegriffe, z.B. Orts- oder Zeitangaben, explizit angeben bzw. auswählen kann. Die Herausforderung für die semantische Suchmaschine besteht darin, die Semantik der Suchanfrage zu erkennen, ggf. unter Zuhilfenahme von Kontextinformationen, und sie auf die Semantik(en) der angeschlossenen Zielsysteme abzubilden.

Die Ergebnispräsentation soll in einem Webportal erfolgen. Dazu sollen grundsätzlich generische Komponenten zum Einsatz kommen. Der Anschluss neuer Zielsysteme soll also nicht automatisch die Entwicklung neuer Frontend-Komponenten bedeuten, vielmehr sollen vorhandene Komponenten automatisch oder maximal durch Konfiguration in der Lage sein, Daten aus dem neuen Zielsystem in der Ergebnispräsentation zur Anzeige zu bringen. Wenn die Ergebnispräsentation aus mehreren Komponenten be-

steht, muss sie dennoch eine konsistente Gesamtsicht auf die Suchergebnisse bieten. Daher müssen (hoch-)konfigurierbare Komponenten zur generischen Darstellung verschiedener Datentypen (Volltext, Messdaten, Objektdaten, Metadaten, Dokumente, Medien) bereitgestellt werden.
Im Idealfall stehen Beschreibungen der Semantik der darzustellenden Daten, z.B. in Form von formalen Datenschemata (Brickley und Guha 2014; schema.org 2016b; jsonschema-org 2018; W3C 2012b), zur Verfügung, welche die Konfiguration von Anzeigekomponenten automatisieren oder zumindest unterstützen können.

Die Orchestrierung der einzelnen Komponenten muss möglich sein, um eine Gesamtanwendung als Zusammenspiel mehrerer Komponenten erstellen zu können. Änderungen in einer Komponente, z.B. die Aktualisierung der Elemente (Inhalte) in einer Liste, muss den anderen Komponenten mitgeteilt werden, damit sie ggf. auf die Veränderung reagieren können.
Bei Nutzerinteraktionen, z.B. dem Verschieben eines Kartenausschnitts oder der Selektion eines Objektes, soll die Konsistenz der Ergebnispräsentation gewahrt bleiben, z.B.

- fehlende Daten nachgeladen werden
- mehrere unterschiedliche Repräsentationen des selektierten Objektes (z.B. in einer Liste, einer Karte und einer Detailansicht) ebenfalls als ausgewählt dargestellt werden
- Änderungen der Suchanfrage zur Aktualisierung der Ergebnispräsentation und/oder ihrer Bestandteile führen.

Das entspricht den Grundzügen moderner Einzelseiten-Webanwendungen („single page Web applications“) (Mikowski et al. 2014; blak-it.com 2017), einer logischen Umsetzungsvariante von Frontends einer serviceorientierten Architektur (Erl 2016; Mesbah und van Deursen 2007), die aus einer einzelnen HTML-Seite besteht und deren Daten bei Bedarf dynamisch (nach-)geladen werden.

Zur Bewertung der Qualität einer (semantischen) Suchemaschine können eine ganze Reihe von Kriterien angelegt werden. Dabei spielen auch die Sichtweisen der beteiligten Personen und Institutionen eine wesentliche Rolle. Für den Nutzer einer Suchmaschine wird immer im Vordergrund stehen, ob seine (mehr oder weniger spezifische) Frage beantwortet wird oder nicht, daneben die möglichst intuitive Benutzung der Suchmaschine, z.B. die einfache, dennoch spezifische Formulierung von Suchanfragen, die intuitive Gestaltung der Benutzeroberfläche oder die übersichtliche Darstellung der Suchergebnisse. Auch der Betreiber einer Suchmaschine sollte das Ziel haben, dass die Fragen der Nutzer hinreichend korrekt beantwortet werden, er muss jedoch auch weitere Aspekte, z.B. den Aufwand zur Erschließung und Aufbereitung von Daten für die Suche sowie z.B. Betriebsaspekte wie die Verfügbarkeit und die Antwortzeit der Suchmaschine im Auge behalten.

Bei einer domänenspezifischen Suchmaschine, die z.B. Daten aus dem Bereich „Umwelt“ bereitstellt, stellt sich zunächst die Frage nach den Grenzen der Domäne, die im

Wesentlichen nach fachlichen Kriterien festgelegt werden müssen. Daneben spielt zur Definition des möglichen Suchraumes auch der Kreis der möglichen Datenbereitsteller eine Rolle, z.B. wenn nur behördlich erhobene und bereitgestellte Daten bzw. Daten mit gesicherter Qualität gefunden werden sollen. Danach stellt sich die Frage nach der Verfügbarkeit und Bereitstellung aller für die Domäne relevanten Daten, die ebenfalls fachlich, jedoch häufig auch nach organisatorischen oder rechtlichen Kriterien erfolgen. Beispielsweise ist zu klären, ob die Zielgruppe „Allgemeine Öffentlichkeit“ auf alle grundsätzlich verfügbaren Daten auch tatsächlich zugreifen darf, oder ob z.B. Datenschutzbestimmungen der allgemeinen Bereitstellung der Daten oder von Teilen der Datensätze widersprechen, z.B. wenn dort personenbezogene Daten enthalten sind.

Wenn die Datenquellen und damit auch der mögliche Suchraum geklärt sind und ein Nutzer eine Suche tatsächlich durchführen kann, stellt sich die Frage, ob er anhand des präsentierten Ergebnisses seine Frage tatsächlich beantwortet bekommt – oder nicht. Dazu ist eine Interpretation des Suchergebnisses notwendig. Die Suchanfrage kann sehr spezifisch sein,, z.B. „Wie hoch war die Feinstaubkonzentration in der Reinhold-Frank-Straße in Karlsruhe am 21.06.2014 um 14:30 Uhr?“[4], jedoch auch sehr weit gefasst und unspezifisch, z.B. „bauen in karlsruhe-knielingen“ – letzteres Muster ist die überwiegende Zahl der Nutzer von Internet-Suchmaschinen gewohnt (Eberspächer und Holtel 2007).

Auch die Präsentation der Suchergebnisse sollte spezifisch erfolgen – d.h. je nach Datentyp sollten die Daten passend präsentiert werden, z.B. Geoobjekte in einer Kartensicht, Messwerte als Tabelle oder Diagramm, Volltexttreffer inklusive der Fundstelle (Snippets) etc. Dabei ist es wichtig, dass die besten Treffer nicht in einer Masse von „Rauschen“, d.h. für die Suchanfrage nicht oder wenig relevanten Informationen, untergehen. Bei wenig spezifischen Suchanfragen kann das eine Gratwanderung sein, einerseits sollen Informationen zu einem unspezifischen Thema wie „Bauen“ gefunden werden, wer sich so einen Überblick verschaffen möchte, darf sich nicht in zu vielen Details verlieren.

Zur Beurteilung der Qualität der Treffer zu einer gegebenen Suchanfrage ist nicht deren Anzahl, sondern in erster Linie die Relevanz der Ergebnisse für den Nutzer entscheidend. Da relevante Treffer im Allgemeinen aus verschiedenen Zielsystemen stammen können, sollten auch Treffer aus allen relevanten Systemen gefunden werden. (Griesbaum et al. 2002) verwenden die Begriffe Makroprecision und Mikroprecision zur Beschreibung der Effektivität einzelner Suchanfragen, im ersten Fall der gesamten Trefferliste, im zweiten Fall des einzelnen Ergebnisses.

---

[4] Google lieferte am 26.06.2016 dazu genau eine Antwort, allerdings nicht den gewünschten Feinstaubwert, sondern eine Meldung von 2015 auf ka-news.de, in der es um Feinstaub in Karlsruhe geht, die gewünschte Straße (dort befindet sich eine Luftmessstation) kommt hier als Beispiel vor. Link: http://www.ka-news.de/region/karlsruhe/Karlsruhe~/Feinstaub-in-Karlsruhe-Atmen-wir-saubere-Luft;art6066,1551668

Die Kriterien zur Bewertung der Relevanz von Suchergebnissen sind daher sowohl für die Trefferliste als Ganzes als auch für die einzelnen Treffer anzugeben, z.B.

- die Vollständigkeit der Treffer
- die Aktualität der Treffer
- die Relevanz bzw. Bewertung der Quelle, z.B. ob es sich um behördliche Informationen handelt
- der Bezug zu weiteren Ergebnissen (Links, Relationen).

(Lewandoski und Höchstötter 2007) führen für die Bewertung von Suchmaschinen vier Evaluationsbereiche an, die sich wiederum aus verschiedenen Evaluationsmaßen zusammensetzen:

- Qualität des Index (Vollständigkeit, Aktualität)
- Qualität der Suchresultate (im Vergleich zu anderen Suchmaschinen bzw. im Vergleich zur absolut zur Verfügung stehenden Information[5])
- Qualität der Suchfunktion (z.B. Möglichkeiten zur Filterung nach Metaattributen wie Sprache oder Dokumenttyp)
- Nutzerfreundlichkeit (Usability).

Für die vorliegende Arbeit sind aufgrund der Vorgaben (gegebene Menge von Zielsystemen, single search slot) vor allem die Qualität der Suchresultate und die Nutzerfreundlichkeit (insbesondere auch bei der Präsentation der Resultate) von Bedeutung.

Da die Bewertung der Suchresultate sehr stark von den Erwartungen des einzelnen Nutzers abhängt, soll die Evaluation anhand ausgewählter Anwendungsszenarien vorgenommen werden.

1.3 Ziele und Aufgaben

Das Ziel der vorliegenden Arbeit besteht darin, ein neues Konzept für die semantische Suche in heterogenen Informationssystemen zu Fragestellungen der Umwelt und Energie zu entwickeln. Dazu sind die folgenden wissenschaftlichen Zielstellungen zu untersuchen:

1. Entwicklung einer **Grundarchitektur** für eine semantische Suche in heterogenen Informationssystemen unter Berücksichtigung von gegebenen technischen und inhaltlichen Randbedingungen einer bestehenden Landschaft von Informationssystemen am Beispiel des Umweltinformationssystems Baden-Württemberg (UIS BW).
   Die Grundarchitektur umfasst die Zielsysteme, in denen gesucht werden soll, Hintergrundwissen in Form von Vokabularen und Gazetteer-Dienste und

[5] Eher ein theoretischer Fall, denn dafür sind Metainformationen notwendig, deren Vollständigkeit kaum geprüft bzw. belegt werden kann.

Webportale zur Präsentation der Suchergebnisse. Der Raum dazwischen bietet viele Freiheiten für eine Grundarchitektur und Ausprägungen davon.

2. Entwurf verschiedener **Architekturvarianten** entsprechend der Grundarchitektur. Die Varianten sollen dabei evolutionsartig aufeinander aufbauen und gleichzeitig Rückflüsse in neue Varianten sowie in Anforderungen und Empfehlungen zur Erweiterung des UIS BW liefern.
3. Vereinheitlichung der **Repräsentation bzw. Abbildung von Semantik** der Daten aus verschiedenen, heterogenen Zielsystemen. Hier sollen ebenfalls verschiedenen Varianten implementiert und evaluiert werden.
   Hierzu müssen Daten ggf. in alternativen Datenformaten bzw. Systemen bereitgestellt werden. Dazu bedarf es der Automatisierung von Datenflüssen, insbesondere zur Wahrung der Aktualität und Datenkonsistenz.
   Die Einbeziehung von Kontextinformationen in die Suchanfrage soll ermöglicht werden.
4. **Erprobung aller Architekturvarianten** anhand repräsentativer Beispiele. Dazu sollen Anwendungsfälle repräsentativer Use-Cases definiert und an den Implementierungen der Architekturvarianten überprüft werden. Da hierfür die Einbindung einer großen Menge „echter“ Daten (nicht synthetischer Testdaten) sinnvoll ist, sollen ebenfalls Daten aus dem UIS BW verwendet werden.
5. **Ableitung von Aussagen zur Leistungsfähigkeit** des neuen Konzeptes aus den Erkenntnissen der Erprobung.

Wie bereits in Abschnitt 1.1 erwähnt wurde, gibt es Randbedingungen, die das Erstellen eines (Domänen-) spezifischen Recherche- und Informationsportals gegenüber der globalen Internetsuche vereinfachen.

Die Anforderungen und Ziele entsprechen in vielen Punkten den Anforderungen an die Suche innerhalb geschlossener Organisationen, z.B. Firmen-Intranets, die häufig unter dem Begriff “Enterprise Search” (Lange 2009) subsumiert werden. Auch hier sind die Datenquellen in der Regel bekannt und gut beschrieben. Im Gegensatz zu vielen Intranet-Systemen sind die jetzt betrachteten Datenquellen jedoch heterogener, insbesondere werden (nicht harmonisierte) Datenquellen externer Anbieter herangezogen, so dass die Bandbreite von wohlbekannten und wohldefinierten Datenquellen (Semantik und Syntax bekannt) bis hin zu „fremden“ Datenquellen (Semantik weitgehend unbekannt, Daten semi- oder unstrukturiert) reicht.

1.3.1 Harmonisierung von Semantik

Im Gegensatz zu Internet-Suchmaschinen stehen für viele Informationssysteme zusätzliche (Meta-)Informationen zur Verfügung. Viele Datenquellen sind bekannt, enthalten strukturierte oder semi-strukturierte Daten und anhand von Metadatenbeschreibungen ist häufig auch die Semantik der Daten (bzw. ihrer Schemata) bekannt. Auf den ersten Blick scheint die semantische Suche in bekannten Datenquellen also einfacher als bei einer allgemeinen Internet-Suche, jedoch besteht auch hier einer Herausforderung in der Harmonisierung der Daten aus verschiedenen Quellen, mit anderen Wor-

ten: Der Übersetzung der Suchanfrage in die jeweilige Semantik (und deren technische Repräsentation) der einzelnen Informationssysteme, die Interpretation der Ergebnisse entsprechend der zugehörigen Semantik und das Zusammenführen der Daten in eine einheitliche Repräsentation der Suchergebnisse (Ergebnis-Mashup) für den Nutzer. Hierbei können semantisch zusammengehörige Daten aus verschiedenen Datenquellen zusammengeführt (z.B. Entfernung von Duplikaten, Ergänzung von Attributen) und einheitlich dargestellt werden.

1.3.2 Umgang mit unterschiedlichen Datentypen

Eine weitere Herausforderung ist die Verschiedenartigkeit von Daten. Unstrukturierte Daten aus Text-Dokumenten, Messdaten, tabellarische Daten, Geodaten etc. liegen in teilweise sehr unterschiedlichen Formaten bzw. Repräsentationen vor und können auf vielfältige, jeweils für den Typ der Daten angemessene, Weise dargestellt werden. Objekte mit Geobezug (z.B. Umspannwerke, Windkraftanlagen oder Schutzgebiete) können beispielsweise in einer Liste (zur Übersicht), einer Tabelle (zusätzlich mit Attributen) oder auf einer Karte dargestellt werden, unter Umständen sogar gleichzeitig in mehreren Formen. Eine Verknüpfung zusammengehöriger Daten oder Daten in verschiedenen Darstellungen bzw. Repräsentationen im Frontend soll möglich sein. So soll das Auswählen eines bestimmten Objektes durch den Nutzer in einer der Ansichten beispielsweise dazu führen, dass das Objekt in allen Ansichten als markiert (ausgewählt) dargestellt wird.

1.3.3 Datenquellen, Datenfluss, Konsistenz

Eine weitere Nebenbedingung, die sich aus der Rolle eines Recherche- und Informationsportals ergibt, ist, dass auf Informationen im Wesentlichen lesend zugegriffen wird. Der unidirektionale Informationsfluss von einem Informationssystem in Richtung des Portals vereinfacht grundsätzlich den Zugriff, dennoch sind für jede Datenquelle Konsistenzbedingungen festzulegen und zu implementieren, insbesondere wenn Daten zwischengespeichert, gecached oder weiterverarbeitet werden. Die Konsistenzbedingungen sollen sicherstellen, dass im Rechercheportal und der originären Datenquelle zu jedem Zeitpunkt konsistente Sichten auf die Daten verfügbar sind bzw. Abweichungen innerhalb eines akzeptablen Bereiches, z.B. eines definierten maximalen zeitlichen Versatzes (Latenz), bleiben.

1.3.4 Nutzung von Standards

Gerade wenn von heterogenen Informationssystemen, dem Mapping von Datenmodellen, der Harmonisierung von Daten und derer Semantik, von generischen Komponenten und von der Interoperabilität von Systemen die Rede ist, sollte es selbstverständlich sein, dass alle (neuen) Datenrepräsentationen und -formate sowie alle definierten Schnittstellen möglichst existierende Standards nutzen.

Bei der Auswahl von Standards ist in der Realität häufig ein gewisser Pragmatismus gefragt, da die Standardisierung von Schnittstellen und Formaten oftmals sehr lange dauert und häufig zu zwar mächtigen, jedoch auch entsprechend schwerfälligen Ergebnissen führt. Als Beispiel seien hier der Standard „Sensor Observation Service“ (SOS) des Open Geospatial Consortium (OGC) genannt, der bereits seit 2007 existiert, jedoch seither eher in akademischen Projekten Anwendung findet und von vielen Sensor-Herstellern nicht unterstützt wird. Stattdessen verwenden viele Hersteller nach wie vor proprietäre, leichtgewichtigere Formate, die sich dennoch zu de-Facto-Standards – teilweise in einem Teilgebiet – entwickelt haben, z.B. memosens (memosens.org 2017) oder MMS Alliance (MSS Alliance 2015). In der vorliegenden Arbeit soll der Begriff des Standards in einer weiter gefassten Definition behandelt werden, die (verbreitete) De-facto-Standards einschließt.

1.3.5 Freie Softwarekomponenten und Nutzung von Open Source

In einem Softwareprojekt wird sich immer wieder die Frage nach der Nutzung vorhandener Lösungen (Komponenten, Bibliotheken) stellen. Häufig gestellte nichtfunktionale Anforderungen an solche Komponenten sind deren kostenfreie Verfügbarkeit bzw. deren Vorliegen als Open Source Software. Die häufig kolportierten Vorteile von Open Source Lösungen (Kostenfreiheit/Lizenzmodell, einfache Verfügbarkeit, Anpassbarkeit, Anbieterunabhängigkeit, Stabilität und Sicherheit) sind jedoch im Einzelfall zu prüfen und gegen eventuelle Nachteile (Fehlen von Service Level Agreements bzw. professionellem Support, Verbot der kommerziellen Nutzung, offene Haftungs- und Gewährleistungsfragen) abzuwägen.

In Rahmen der vorliegenden Arbeit sollen freie und Open Source-Komponenten bevorzugt werden.

1.3.6 Abgrenzung

In der vorliegenden Arbeit sollen auch Konzepte und Technologien aus dem Bereichen der Wissensrepräsentation und des Semantic Web genutzt werden, es geht jedoch explizit nicht darum einzelne Informationssysteme für das Semantic Web (Web 3.0) aufzurüsten.

Die Wissensrepräsentation (Knowledge Representation) setzt sich dabei im Wesentlichen aus den Bereichen Logik (Strukturen zur Bildung von Regeln, die zum Schließen genutzt werden können), Ontologien (Definition von Konzepten zur Repräsentation von Objekten und Beziehungen dazwischen) und der Berechenbarkeit zusammen. Im Gegensatz zum bestehenden WWW der Dokumente wird das Semantic Web also einzelne Objekte enthalten, die eindeutig semantisch beschrieben und adressierbar sind. Grundvoraussetzungen hierfür sind Linked Data, Vokabulare, Mechanismen zum Abfragen (Queries) (s. Anhang A1.3) und Inferenzen (s. Anhang A1.4).

Wie oben beschrieben, können die Ideen und Prinzipien des Sematic Web als Richtschnur für Konzepte und Entwicklungen dienen, insbesondere um auch für die be-

schriebene semantische Suche von den zu erwartenden Entwicklungen in Richtung eines Web 3.0 profitieren zu können.

In der vorliegenden Arbeit geht es weiterhin nicht um die Entwicklung eines domänenspezifischen Vokabulars bzw. einer Domänen-Ontologie[6]. Vielmehr sollen existierende Vokabulare bzw. Ontologien, auch aus an den Domänen „Umwelt“ und „Energie“ angrenzenden Bereichen, genutzt und miteinander vernetzt bzw. entsprechende Mechanismen aufgezeigt, entwickelt und angewendet werden.

Es geht auch nicht darum, massenhaft Daten(-sätze) aus existierenden Informationssystemen als Instanzen in bestehende Ontologien zu integrieren, sondern vielmehr Klassen von Daten mit Hilfe von Vokabularen zu beschreiben und so eine semantisch eindeutige Einordnung von Daten aus verschiedenen Datenbeständen zu erreichen.

Es existieren eine ganze Reihe semantischer Suchmaschinen[7], die den Anspruch haben, dem Nutzer auf einer Ergebnisseite eine konkrete Antwort zu einer bestimmten Frage zu geben. Jede Antwort wird auf Basis des vorhandenen Datenbestands (in einer sehr großen Datenbank) berechnet, ggf. als mathematisch korrektes Ergebnis der Anfrage „Wo befindet sich gerade die ISS?“ bei der semantischen Suchmaschine „Wolfram|Alpha“ (Wolfram|Alpha 2017). In der vorliegenden Arbeit geht es nicht um die Entwicklung einer solchen semantischen Suchmaschine, dazu müssten die Daten wie oben beschrieben vollständig in eine entsprechende Datenbasis, z.B. Ontologie, überführt werden.

1.3.7 Was ist neu an der vorliegenden Arbeit?

Es gibt eine Vielzahl von (Internet-)Suchmaschinen, meistens textvergleichbasiert, aber auch einige mit semantischer Unterstützung oder vollständig semantischer Erfassung von Suchanfragen. Für Internet-Suchmaschinen ist eine semantische Suche

---

[6] Hier existieren Entwicklungen verschiedener Wissenschaftler und Forschungsgruppen, z.B. ONTOENERGY (Linnenberg et al. 2013) oder OpenWatt (Lamanna/ Maccioni, 2014); Joint Thesaurus der IAEA (http://www.etde.org/edb/JRS1r1_web.pdf), WAND Electric and Gas Utility Taxonomy, Urban Energy Ontology (http://www.semanco-tools.eu/urban-enery-ontology)

[7] Beispiele für semantische Suchmaschinen:

- **AskWiki**: Semantische Suchmaschine für den Datenbestand der deutschsprachigen Wikipedia mit Eingabe der Anfrage per Sprache
- **GoPubMed**: Semantische Suchmaschine für die biomedizinische Domäne. http://www.gopubmed.com/web/gopubmed/
- **Swoogle**: Semantische Suchmaschine, die Dokumente, Begriffe und Daten im semantischen Web suchen kann; http://swoogle.umbc.edu/
- **WolframAlpha**: „Antwortmaschine“ des Mathematikers Stephen Wolfram mit Schwerpunkt auf den exakten Wissenschaften; http://www.wolframalpha.com/

(Verstehen der Suchanfrage und liefern einer spezifischen Antwort) prinzipiell schwierig, da sie bei Suchanfragen mit der gesamten natürlichen Sprache und deren vollständigem Wortgut umgehen müssen. Zum Beispiel erschweren Mehrdeutigkeiten (Homonyme), Synonyme, Flexionen und Zusammensetzungen (insbesondere in der deutschen Sprache) das Erkennen der (möglichst eindeutigen) Bedeutung einer Suchanfrage – das kann ggf. durch Rückfragen und eine Auswahl durch den Nutzer aufgelöst werden. Für einige der beschriebenen Probleme existieren Lösungsansätze, z.B. hinterlegte Wörterbücher, Synonymlisten, Thesauri etc.

Eine deutliche Einschränkung von Internet-Suchmaschinen stellen jedoch die durch sie erschlossenen Inhalte dar: Die meisten Internet-Suchmaschinen bieten nur Zugang zu solchen Inhalten, die über eine WWW-Repräsentation (HTML-Seiten) verfügen oder per URL als Datei-Download (Dokument) verfügbar sind. Andere Systeme (wie Datenbanken) werden meist nicht durchsucht.
Die Indizierung von HTML-Seiten liefert häufig unscharfe Ergebnisse, da neben den Nutzinhalten weitere Elemente (Navigation, Kopfzeile, Fußleiste, …) in den Seiten enthalten sind, die ebenfalls indexiert werden, jedoch mit dem eigentlichen Nutzinhalt nicht direkt in Beziehung stehen. Bei HTML5-Seiten kann die fehlende inhaltliche Auszeichnung solcher für Suchmaschinen nicht relevanter Bereiche durch den Einsatz entsprechender Tags (menu, footer, summary, nav, …) reduziert werden.

Im Gegensatz zu Internet-Suchmaschinen stehen für die Aufgabenstellung der vorliegenden Arbeit weitergehende Informationen zur Verfügung, die in einer Architektur für die übergreifende Suche in heterogenen Informationssystemen bzw. verschiedenen Ausprägungen einer solchen Architektur, zusammenzuführen sind:

- Die Domäne(n) der Suchanfragen ist/sind beschränkt und bekannt. Es handelt sich um die Domänen „Umwelt“ und „Energie“. Damit ist das Hintergrundwissen und die Anzahl der ihm zugrundeliegenden Vokabulare beschränkt. Sie lassen sich daher auch in praktischen Projekten für die Suche verwenden.
- Ebenso sind die angeschlossenen heterogenen Datenquellen bekannt und in ihrer Anzahl beschränkt. Sie enthalten strukturierte oder semi-strukturierte Daten und anhand von Metadatenbeschreibungen ist häufig auch die Semantik der einzelnen Daten (bzw. Schemata/Datenquellen) bekannt.
- Auf den ersten Blick scheint die Suche in solchen Datenquellen also einfacher, jedoch besteht die Herausforderung in der Harmonisierung der Daten aus den verschiedenen Quellen, mit anderen Worten: Der Übersetzung der Suchanfrage in die jeweilige Semantik (und in die technische Repräsentation der Anfrage) der einzelnen Zielsysteme, die Interpretation der Ergebnisse (nach der Semantik) und die Zusammenführung der Daten in einer einheitlichen Repräsentation der Suchergebnisse (Ergebnis-Mashup).
- Hierbei können zusammengehörige Daten aus verschiedenen Datenquellen zusammengeführt (z.B. Entfernung von Dubletten, Ergänzung von Attributen) und einheitlich mit Hilfe von generischen Komponenten dargestellt werden.

- Eine wesentliche Herausforderung ist dabei auch die Verschiedenartigkeit von Daten(-typen). Unstrukturierte Daten aus Text-Dokumenten, Messdaten, tabellarische Daten, Geodaten etc. können auf vielfältige, jeweils für den Typ der Daten angemessene, Weise dargestellt werden. Eine Verknüpfung zusammengehöriger Daten oder Daten in verschiedenen Darstellungen soll dabei möglich sein.

Das beschriebene Vorgehen ist neu für das Umweltinformationssystem Baden-Württemberg, in welchem bisher inselartige Suchlösungen (für einzelne Systeme) oder textvergleichbasierte Suchlösungen, z.B. im Umweltportal, verwendet werden, die jedoch im Wesentlichen auf die Suche in Websystemen und Dokumentenbeständen beschränkt sind.

„Klassische“ Firmen-Suchlösungen bieten meist wenig Integration der verschiedenen Datenquellen an. Entweder werden alle Datenquellen in einer einzelnen Trefferliste angezeigt, oder – je nach Datentyp – in verschiedenen Trefferlisten (nebeneinander) dargestellt. Ein Bezug zwischen einzelnen Treffern in verschiedenen Listen wird meist nicht hergestellt. Je nach Typ der Suchmaschine werden viele Daten dabei nicht attributweise dargestellt, sondern in einer einheitlichen Form (Titel, Link und „Snippet“) präsentiert. Eine Datentyp-spezifische Anzeige der Daten ist „out of the box“ mit den wenigsten Suchmaschinen möglich.

Im Gegensatz zu vielen kleinen bis mittleren Firmen-Intranets sind die hier verwendeten **Datenquellen jedoch heterogener**, insbesondere werden (nicht harmonisierte) Datenquellen externer Anbieter herangezogen, so dass die Bandbreite zwischen wohlbekannten und wohldefinierten Datenquellen (Semantik und Syntax bekannt, strukturierte Daten) bis hin zu „fremden“ Datenquellen (Einstiegspunkt/Schnittstelle bekannt, Semantik weitgehend unbekannt, semi- oder unstrukturierte Daten) geht.

1.4 Übersicht über die Arbeit

Im folgenden Kapitel 2 werden grundlegende Prinzipien für die Architektur einer neuartigen semantischen Suche (für die Domänen Umwelt und Energie) entwickelt. Dabei werden bewusst Freiheitsgrade gegeben, die verschiedene Umsetzungen nach den entwickelten Prinzipien zulassen. Dadurch soll eine „Evolution“ der semantischen Suche ermöglicht werden, die eine Evaluierung der konkreten Umsetzungsvarianten in möglichst realen Umgebungen ermöglichen soll. Startpunkt der Evolution ist ein Suchportal mit einer klassischen Volltextsuche – im Gegensatz zu einer semantischen Suche.

In Kapitel 3 werden verschiedene Varianten der Umsetzung der Architektur vorgestellt. Dabei kommen unterschiedliche Technologien zum Einsatz, die der angesprochenen technischen Evolution Rechnung tragen. Die Ergebnisse der Evaluierung jeder Evolutionsstufe fließen jeweils in die folgenden Evolutionsstufen ein.

Kapitel 4 illustriert die verschiedenen Evolutionsstufen anhand konkreter Umsetzungsbeispiele, die teilweise produktiv im Einsatz sind oder waren. Das soll den Anspruch der vorliegenden Arbeit betonen, eine praxistaugliche Architektur und tatsächlich nutzbare Systeme zu entwickeln.

Die Gegenüberstellung, Diskussion und Bewertung der beschriebenen Architekturvarianten ist in Kapitel 5 beschrieben. Kapitel 6 zieht ein abschließendes Fazit und bietet einen Ausblick für weitere Entwicklungen.

2 Ein neues Konzept für die semantische Suche

Im zweiten Kapitel soll ein neues, allgemeines Konzept für die semantische Suche in heterogenen Informationssystemen entwickelt werden. Das allgemeine Grundkonzept wird anschließend in verschiedenen Architekturvarianten umgesetzt werden, die sich im Einsatz verschiedener Technologien zur Realisierung architektonischer Grundaufgaben unterscheiden.

2.1 Grundidee und Übersicht

Abbildung 2 zeigt die wesentlichen Komponenten des neuen Konzeptes für die semantische Suche.

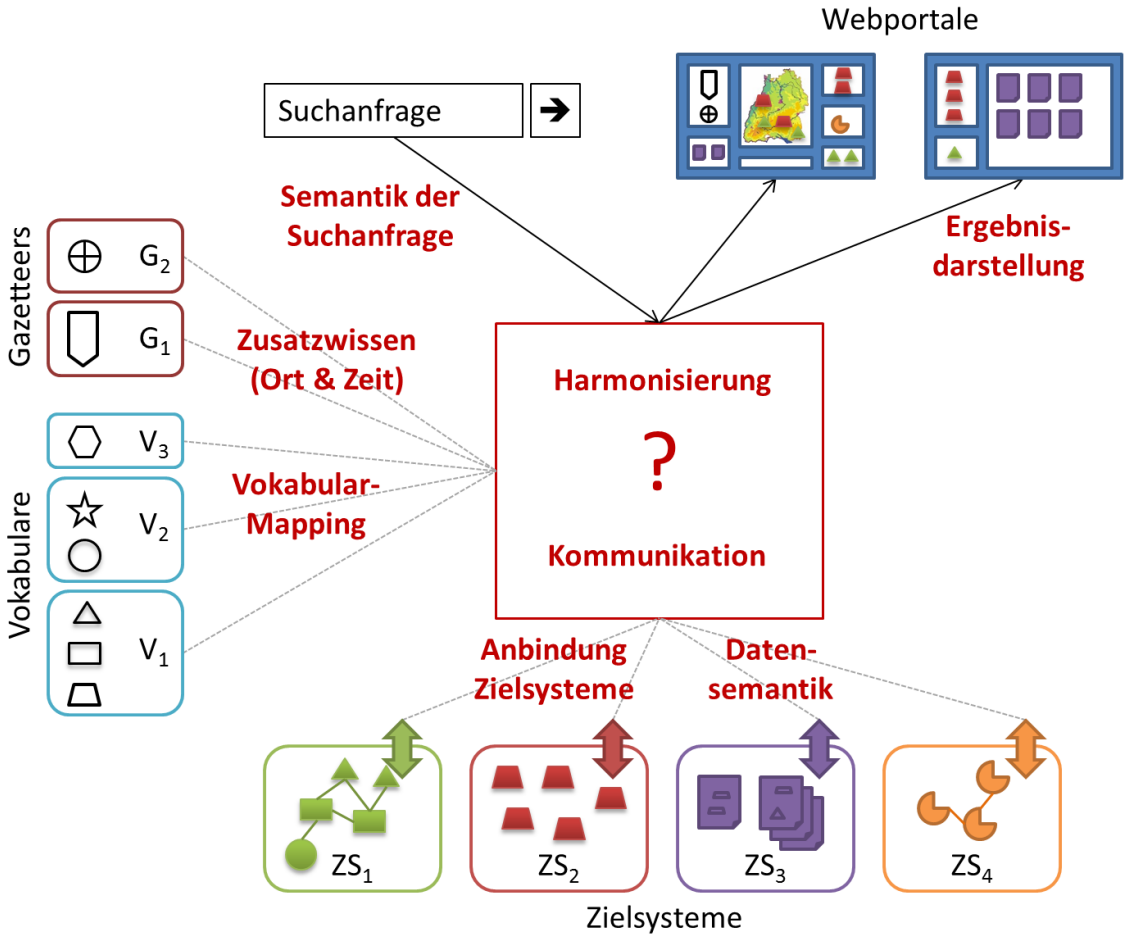

Abbildung 2: Rahmen einer allgemeinen Architektur für die semantische Suche

Als gegeben werden eine Anzahl heterogener Informationssysteme (Zielsysteme ZS_i), eine Anzahl von Vokabularen V_i sowie eine Anzahl von Diensten, die Zusatzwissen bereitstellen können (hier: „Gazetters“ G_i), angenommen. Die nutzende Anwendung besteht dabei aus einer Oberfläche zur Formulierung von Suchanfrage (vereinfacht: „Suchschlitz“) sowie Komponenten zur Darstellung der Suchergebnisse in Webportalen (Ergebnisdarstellung).

Alle weiteren Komponenten der Grundarchitektur werden zunächst als offen angenommen, d.h. grundsätzlich können beliebige Module, die einen Beitrag zur Erreichung der Ziele leisten können, hinzugefügt werden – selbstverständlich mit dem Ziel, den Gesamtaufwand, d.h. die Anzahl und die Komplexität der Module, möglichst gering zu halten.

Die Grundarchitektur besteht daher aus wenigen zentralen Grundaufgaben:

1. Erfassung der Semantik von Zielsystemen ZS_i
2. Harmonisierung der Semantik verschiedener Zielsysteme (Vokabulare V_i)
3. Realisierung des technischen Zugriffs auf die Zielsysteme ZS_i
4. Klassifikation des Typs von Informationen anhand generischer Datentypen
5. Realisierung der Darstellung von Beziehungen zwischen Daten/Objekten
6. Erfassung der Semantik der Suchanfrage
7. Nutzung von Zusatzinformationen zur Anreicherung bzw. Explizierung der Suchanfrage (z.B. Koordinaten zu Ortsnamen) mit Hilfe der Gezetteer-Dienste G_i
8. Nutzung der Semantik der Suchanfrage zur Abfrage der Zielsysteme
9. Koordination der Ergebnisdarstellung (Generierung eines Mashup)
10. Darstellung von Informationen entsprechend ihrer Klassifikation durch generische Komponenten.

Neben den obligatorischen Grundaufgaben gibt es einige optionale Bestandteile, die zur Erfüllung weitergehender Aufgaben genutzt werden können, z.B. die Bereitstellung von Daten als Linked Data im Sinne des Semantic Web (Berners-Lee 2006):

11. Bereitstellung von Daten über eindeutige Bezeichner (URIs)
12. Darstellung von Beziehungen zwischen Daten (Objekten) mit Hilfe von URIs
13. Kommunikation zwischen Darstellungskomponenten.

Die erste Hauptaufgabe besteht in der Harmonisierung der Semantiken der verschiedenen Zielsysteme ZS_i respektive der ihnen zugrundeliegenden Vokabulare V_i. Das kann z.B. durch die Zusammenfassung zusammengehörender/identischer Konzepte aus den verschiedenen Vokabularen oder über die Abbildung auf ein (zu definierendes) übergreifendes Vokabular geschehen.

Die zweite Hauptaufgabe besteht in der übergreifenden Präsentation der verschiedenartigen Suchergebnisse aus unterschiedlichen Zielsystemen mittels verschiedener generischer Frontend-Komponenten sowie in der Orchestrierung der Komponenten zur Erzeugung einer stimmigen Ergebnispräsentation. Hierfür ist eine Kommunikation zwischen den Komponenten notwendig.

Die verschiedenen Bausteine des neuen Konzeptes werden im Folgenden beschrieben. Aus ihnen ergibt sich eine Gesamtarchitektur mit einer ganzen Reihe von Freiheitsgraden bei der Entwicklung, die zur Implementierung verschiedener konkreter Architekturvarianten führt, die in Kapitel 3 beschrieben werden.

2.2 Zielsysteme

Als Zielsysteme ZS_i kommen alle Informationssysteme infrage, die Daten für die gegebenen Domänen zur Verfügung stellen können. Dabei ist es zunächst weitgehend irrelevant, in welcher Form die Daten vorliegen bzw. welche Schnittstellen zum Zugriff auf die Daten bereitgestellt werden. Dennoch müssen für die Zielsysteme einige Festlegungen getroffen werden.

2.2.1 Definition

Jedes Zielsystem ZS_i stellt genau eine Klasse von Daten (Objekten) zur Verfügung, d.h. alle Daten gehören zu genau einem semantischen Konzept. Das bedeutet, dass Zielsysteme in erster Linie strukturierte Daten mit einer gegebenen Semantik zur Verfügung stellen.

Was für viele Informationssysteme, die im Allgemeinen Objekte verschiedener Konzepte sowie Beziehungen dazwischen enthalten, zunächst eine Einschränkung darstellt, lässt sich jedoch leicht auflösen, denn jedes Informationssystem, das verschiedene Konzepte verwaltet, lässt sich als Menge mehrerer Informationssysteme darstellen, die jeweils nur ein Konzept bereitstellen[8].

Zielsysteme stellen Daten elektronisch per Netzwerk (Internet) erreichbare Schnittstellen zur Verfügung. Dabei ist die Nutzung standardisierter Schnittstellen und Formate zwar hilfreich, jedoch keine obligatorische Anforderung, da der Zugriff z.B. über Adapter (Gamma 2004) oder Fassaden (Gamma 2004) mit entsprechenden Transformationen[9] realisiert werden kann. Der Zugriff auf Zielsysteme kann auch indirekt, z.B. über Indexe oder strukturierte Suchmaschinen stattfinden, wenn die Konsistenz[10] der redundant gehaltenen Daten gewährleistet ist.

2.2.2 Zielsysteme mit un- bzw. schwach strukturierten Inhalten

Als Spezialfall von Zielsystemen können Web- und Dokumentensysteme betrachtet werden, die Informationen in unstrukturierter (Eriksdotter 2009) bzw. schwach strukturierter/semistrukturierter Form (Bry et al. 2001) bereitstellen, z.B. in Form von HTML-Seiten oder Dokumenten (PDF, DOC etc.). Die semantische Klassifikation solcher Daten über eine Metaebene („Dokument“) hinaus ist im Allgemeinen schwierig und Bedarf einer Analyse des Inhalts bzw. der Metadaten des Dokuments. Ein einzelnes Dokument kann dabei im Allgemeinen eine Vielzahl verschiedener Konzepte enthalten oder

---

[8] Beziehungen zwischen Objekten verschiedener Konzepte gehen hierbei möglicherweise verloren und müssen durch geeignete Mechanismen erhalten werden.

[9] Die Implementierung von Adaptern bzw. Fassaden erzeugt Aufwand, der im Einzelfall erheblich sein kann. Zur Abbildung der Semantik eines Systems können Transformationen jedoch ohnehin notwendig sein.

[10] Der Konsistenzbegriff ist für jedes Zielsystem zu definieren.

Beziehungen zu Konzepten bzw. Objekten haben. Dennoch lassen sich auch Dokumente klassifizieren, indem ihnen Beziehungen zu bekannten Konzepten zugordnet werden. Dabei können sowohl der Inhalt als auch Metainformationen genutzt werden, z.B.

- „Dokument X“ – „befasst-sich-mit“ – „Windkraftanlage“,
- „Dokument X“ – „ist-ein“ – „Forschungspapier“,
- „Dokument X“ – „enthält-Verweis-auf“ – „Dokument Y“.

Es gibt eine ganze Reihe von Diensten, die eine automatisierte Klassifizierung von Dokumenten für ein gegebenes Vokabular übernehmen können[11].

Un- und schwach strukturierte Daten können alternativ, ohne Erfassung ihrer Semantik, auch über „klassische“ Suchmaschinentechnologie, d.h. durch Verwendung der originalen Suchanfrage, bereitgestellt werden.

2.2.3 Semantik von Zielsystemen

Jedes Zielsystem kann für die Darstellung seiner Daten (Objekte) eine eigene Semantik enthalten, die sich z.B. in der Namensgebung, der Auswahl von Attributen, der Verwendung bestimmter Formate und Einheiten, der Verwendung von Ober- und Unterkonzepten etc. ausdrückt.

In vielen Fällen liegt die Semantik nicht explizit und in durch Maschinen prozessierbarer Form vor, sondern steckt direkt in den Daten oder in Metainformationen, z.B. in beschreibenden Dokumenten (Holten 1999).

Um eine Harmonisierung der Semantik verschiedener Zielsysteme erreichen zu können, muss die Semantik jedes einzelnen Zielsystems explizit erfasst werden, um auf die Semantik der anderen Zielsysteme abgebildet werden zu können.

Die Einschränkung, dass Zielsysteme so aufgefasst werden, dass sie jeweils nur Daten eines einzelnen Konzeptes enthalten dürfen, vereinfacht die Beschreibung ihrer Semantik.

2.2.4 Generische Datentypen

Da Daten aus einer Vielzahl von Zielsystemen in einer integrierten Ergebnisdarstellung präsentiert werden sollen, ist es notwendig, deren Darstellungen generisch zu halten.

Im Laufe der Arbeit wurden für die Bereiche Umwelt und Energie im Wesentlichen sechs Klassen von Datentypen (s. Anhang A2) identifiziert, die sich generisch behandeln lassen:

- Objektinformationen

---

[11] Zum Beispiel der Semantic Network Service (SNS) des Umweltbundesamtes. http://www.semantic-network.de (Umweltbundesamt 2016)

    - Einzelne Objekte
    - Listen von Objekten
- Geodaten bzw. Daten mit explizitem Geobezug (beinhalten in der Regel Objektinformationen)
    - Einzelne Objekte
    - Mengen von Objekten (z.B. Kartenlayer)
- Zeitreihen bzw. Messdaten (mit einem Zeitbezug einzelner Messwerte)
- Tabellarische Informationen[12]
- Metadaten, in verschiedenen Granularitäten und Ausprägungen[13]
- Binäre Assets (inkl. zugehöriger Metadaten)
    - Durchsuchbare Assets (Dokumente)
    - Mediendateien (Bilder, Videos, Audios).

Die Daten eines einzelnen Zielsystems können dabei gleichzeitig zu mehreren Klassen gehören, d.h. es kann verschiedene Schnittstellen geben, die verschiedene Sichten auf Daten erlauben, z.B. Windkraftanlagen als Objekte, als Objekte mit Geobezug und als tabellarische Informationen.

Wenn Ergebnisse einer semantischen Suche präsentiert werden sollen (Abschnitt 2.5) ist eine möglichst einheitliche Behandlung von Daten innerhalb der Datentyp-Klassen sinnvoll. Das bedeutet, dass entweder die Komponente zur Anzeige einer Datentyp-Klasse die Datenformate aller möglichen Zielsysteme verstehen muss oder dass die Zielsysteme mindestens ein von der Anzeigekomponente unterstütztes Format[14] liefern können muss. Um die Komplexität der Anzeigekomponenten nicht zu groß werden zu lassen und um notwendige Änderungen an bestehenden Zielsystemen zu vermeiden, ist hier evtl. ein Mittelweg notwendig, z.B. das Bereitstellten von Daten über Adapter bzw. Fassaden, die den Zugriff mittels standardisierter Schnittstellen und Formate ermöglichen. Das kann auch über redundante Systeme geschehen, z.B. Indexe.

2.3 Vorverarbeitung der Suchanfrage

Suchanfragen lassen sich wie Dokumente gegen ein vorhandenes Vokabular klassifizieren und so auf die darin enthaltenen Konzepte bzw. Deskriptoren abbilden.

Die so gewonnenen Informationen können verwendet werden, um die Suchanfrage anzureichern, z.B. der Suchanfrage weitere Attribut-Wert-Paare oder Strukturinformationen (z.B. Oberbegriffe, Synonyme) hinzuzufügen.

---

[12] Semantisch gibt es hier Überschneidungen mit (Listen von) Objektinformationen, z.B. Resultate von DB-Abfragen.

[13] Je nach Kontext können z.B. auch Suchergebnisse einer Volltextsuchmaschine als Metadaten betrachtet werden.

[14] Inklusive Unterstützung der Anfrage-Schnittstelle.

Ein Spezialfall davon ist die Explizierung von Suchbegriffen. Beispielsweise kann der Suchbegriff „Karlsruhe“ zunächst als Ortsangebe (Ortsname, Landkreis, Regierungsbezirk) erkannt und anschließend um Attribute wie einen Mittelpunkt (Längen- und Breitengrad) oder einen Gemeindeschlüssel ergänzt werden, die dann zur Abfrage bzw. Adressierung von Zielsystemen zur Verfügung stehen.

Orts- und Zeitangaben nehmen hier eine Sonderstellung ein. Sehr viele Objekte haben einen Ortsbezug (Standort, ggf. zeitlich veränderlich[15]). Messwerte und Zeitreihen haben einen Zeitbezug. Insofern stellen Ort und Zeit universelle Größen dar, bezüglich derer Objekte miteinander in Beziehung gebracht werden können („ist in der Nähe von“, „ist innerhalb von“, „ereignet sich zur selben Zeit“ etc.), ohne dass eine Beziehung zunächst explizit vorliegt. Gerade bei der Suche nach konkreten Objekten stellt der Ortsbezug einen gängigen Anwendungsfall dar.

2.4 Abbildung und Harmonisierung von Vokabularen

Zielsysteme verwenden im Allgemeinen jeweils eine eigene Semantik zur Darstellung ihrer Daten. In gutartigen Fällen ist die Semantik explizit definiert und verwendet vorliegende Vokabulare bzw. Schemata[16]. Leider werden dabei in den wenigsten Fällen global abgestimmte Vokabulare, wie sie z.B. durch schema.org (schema.org 2016a) vorangetrieben werden, verwendet, sondern es kommen z.B. „nur“ projekt- oder behördenweit abgestimmte bzw. proprietäre Vokabulare zum Einsatz.

Um dennoch eine einheitliche semantischen Suche realisieren zu können, muss eine übergreifende Nutzung von Vokabularen ermöglicht werden, z.B. durch Abbildung von gleichen/ähnlichen Konzepten aufeinander oder durch die Darstellung von Beziehungen zwischen Konzepten verschiedener Vokabulare („is-a“, „is-in-semantic-relation-with“).

Hinzu kommt die Notwendigkeit zur Beschreibung der Semantik von Systemen, bei denen die Semantik nicht explizit vorliegt, z.B. in Form von zusätzlichen Metadaten.

2.5 Integrierte Ergebnisdarstellung (Mashup)

Zur Erzeugung eines Ergebnis-Mashups ist es meist sinnvoll, Klassen gleichartiger Datentypen gemeinsam darzustellen, z.B. alle Geodaten gemeinsam innerhalb einer Kartenansicht oder alle Fotos zusammen innerhalb einer Slideshow.

Dabei sollte es für den Nutzer einerseits keine Rolle spielen, dass die dargestellten Daten aus unterschiedlichen Quellen kommen, andererseits sollte es ihm auch ermög-

---

[15] Zum Beispiel wechseln mobile Messstationen regelmäßig ihren Standort.

[16] In der vorliegenden Arbeit wird immer von einer Kombination von Konzepten und den zugehörigen Schemata ausgegangen.

licht werden, die dargestellten Daten fachlich zu selektieren bzw. zu filtern, z.B. die Darstellung aller Windkraftanlagen auf der Karte ein- bzw. auszuschalten.

Hinzu kommt, dass dieselben Daten in verschiedenen Ausprägungen betrachtet bzw. dargestellt werden können, z.B. können Windkraftanlagen anhand ihrer Geokoordinaten auf einer Karte dargestellt werden und gleichzeitig eine Liste aller Windkraftanlagen (ggf. beschränkt auf den Geobezug des Kartenausschnitts) mit deren wichtigsten Attributen (Leistung, Rotordurchmesser, Nabenhöhe) präsentiert werden, ggf. ergänzt um die Detailansicht zu einer vom Benutzer ausgewählten einzelnen Anlage. Eine Koordination der möglichen Darstellungsformen (Ergebnis-Mashup) ist also notwendig, darüber hinaus evtl. auch eine Kommunikation zwischen den einzelnen Darstellungen, spätestens sobald der Nutzer mit den Darstellungen interagieren (z.B. einzelne Objekte auswählen) können soll.

2.6 Verbindende Schicht zur Beschreibung und Realisierung von Anwendungen

Zur Erzeugung integrierter Ergebnisdarstellungen (Mashups) bedarf es also der Koordinaten zwischen den einzelnen Darstellungskomponenten bzw. auch der Kommunikation der Komponenten untereinander.

Die Koordination muss berücksichtigen, welche Zielsysteme Daten in welcher Form liefern können, welche Klassen von Datentypen im (gesamten) Suchergebnis vorliegen und ggf. welche Arten der Darstellung überhaupt gewünscht sind, d.h. ob es z.B. personalisierte Einstellungen/Präferenzen des Nutzers gibt oder ob ein Redakteur für einen bestimmten Anwendungsfall bereits Voreinstellungen getroffen hat, z.B. dass eine bestimmte Zusammenstellung von Ansichten für einen Anwendungsfall bereits existiert.

Die Koordination bzw. Kommunikation von/zwischen Komponenten stellt sozusagen den Klebstoff dar, der die Orchestrierung von unabhängigen Komponenten im Sinne einer (Such-)Anwendung erlaubt. Je nach Möglichkeit zum Triggern der Suche (z.B. durch Verwendung von durch einen Redakteur oder Autor vorgegebenen Suchanfragen innerhalb einer Webseite) wird mit Hilfe der beschriebenen Mechanismen ganz allgemein die Präsentation von Daten, und damit der Betrieb von Informationssystemen, ermöglicht.

Im folgenden Kapitel werden verschiedene konkrete Architekturvarianten präsentiert, welche jeweils einige, mehrere oder alle der vorgestellten Bausteine enthalten und die entsprechenden Aufgaben umsetzen. Die verschiedenen Architekturvarianten bauen dabei aufeinander auf, d.h. die in einer Architekturvariante gemachten Erfahrungen gehen in die Entwicklung der folgenden Architekturvarianten ein.
Das Vorgehen ist pragmatisch. Es setzt auf der in den Landesumweltportalen Baden-Württemberg, Sachsen-Anhalt und Thüringen vorhandenen Suchfunktionalität (einer kommerziellen Suchmaschine, Stand 2009) auf, und versucht sie sukzessive zu ver-

bessern, wobei die Komplexität des Gesamtsystems sowie die Menge der über die Suche verfügbar gemachten Daten stetig zunimmt.

3 Architekturvarianten

3.1 Übersicht

In Kapitel 3 werden vier Architekturvarianten in ihrer zeitlichen Entwicklung gegenübergestellt. Die Varianten entstanden in Form von Evolutionsstufen der in Kapitel 2 entwickelten neuen Architektur.

Alle Varianten besitzen gemeinsame Elemente der in Kapitel 2 vorgestellten Grundarchitektur, die jedoch in verschiedenen technischen Ausprägungen bzw. Umsetzungsvarianten zum Einsatz kommen. Sämtliche Varianten werden praktisch umgesetzt und anschließend evaluiert. Insbesondere ihre jeweiligen Nachteile dienen zur Entwicklung möglicher Verbesserungen bzw. zum Ableiten von weiteren Architekturprinzipien für die jeweils folgenden Evolutionsstufen hin zu einer Zielarchitektur.

Die erste Variante, eine Art Bestandsaufnahme des Status Quo, stellt den Ausgangspunkt der Überlegungen dar und offenbart dabei gleich so viele Schwächen, dass eine größere Zahl von Anforderungen und Prinzipien daraus abgeleitet werden können.

3.2 Grundlagen

Zunächst wurde das Konzept für die semantische Suche mittels SearchBroker anhand einer rein serverseitigen Anwendung erstellt (Abecker et al. 2009a; Bügel et al. 2010). Im Laufe der Zeit ergab sich jedoch, vor allem getrieben durch Einführung von HTML5, ein Quantensprung in den clientseitigen Möglichkeiten von Webanwendungen. Viele Probleme, sowohl technische wie die Cross-Origin Policy (W3C 2009a) als auch „weiche“, wie die breite Nutzer-Akzeptanz von JavaScript, waren gelöst, und rund um HTML5 entstanden eine große Menge von Frameworks und Bibliotheken, die eine Verschiebung von Mechanismen vom Server auf den Client, d.h. in den Browser, möglich machten. Funktional reiche HTML5-Anwendungen waren möglich, ebenso wie die Nutzung bzw. der Aufbau serviceorientierter Architekturen (SOA) (s. Anhang „Serviceorientierte Architekturen“) in Informationssystemen, die ebenfalls durch das Aufkommen leichtgewichtiger Technologien wie REST (REpresentational State Transfer, s. Anhang A4.2) oder JSON gefördert wurden (Zustandslose Client-Server-Kommunikation mit identifizierbaren Ressourcen als zentrale Dekompositionseinheit, auf Basis uniformer Schnittstellen, die sich nahtlos in das Konzept „Hypermedia“ einfügt. „Leichtgewichtig“ steht dabei z.B. im Gegensatz zu klassischen Webservices, z.B. „SOAP“).

3.2.1 Server-Zentrierung

Die Konzentration auf eine rein serverseitige Lösung barg also plötzlich erhebliche Nachteile, insbesondere was die Möglichkeiten zum Aufbau von ergonomischen, funk-

tional reichhaltigen und performanten Benutzerschnittstellen (bzw. die „User Experience“) betrifft. Die wesentlichen Nachteile einer serverzentrierten Architektur sind (Wang et al. 2015):

- Der Server wird zum Flaschenhals. Alle Informationen laufen hier zusammen. Synchrone Aufrufe, z.B. zum Laden von Daten aus Datenbanken oder von externen Diensten, verzögern ggf. die Auslieferung der Seite, für den Nutzer entstehen spürbare Reaktionszeiten und Pausen.
- Nutzerinteraktionen müssen in der Regel an einen Server übermittelt und von der serverseitigen Anwendung verarbeitet werden. Kommt es dabei zu Veränderungen in der Darstellung, muss in den meisten Fällen die gesamte Seite neu geladen werden. Viele Routine-Aufgaben, z.B. das Generieren des Rahmenlayouts oder einer Navigationsleiste, werden unnötig häufig, d.h. ohne Veränderung gegenüber dem vorigen Aufruf, ausgeführt.
- Bei jedem Laden der Seite entsteht eine ganze Kette von Zugriffen auf Hintergrund-Dienste wie Datenbanken oder externe Die (Wang et al. 2015)nste, selbst wenn dabei dieselben Abfragen erneut ausgeführt werden. Caching-Mechanismen können zwar zur Reduzierung beitragen, das Grundproblem besteht jedoch grundsätzlich weiter.
- Rein serverseitig generierte Oberflächen bieten in der Regel wenig Komfort für den Nutzer. Hybride UI-Ansätze (Frameworks wie Java Server Faces, JSF) können hier erste Abhilfe schaffen, erfordern jedoch auch Änderungen in der Server-Anwendung, z.B. um das Nachladen von Inhalten beim Blättern in einer Liste oder bei der Änderung der Sortierung zu ermöglichen.

Die Serverzentrierung bietet jedoch auch Vorteile (Wang et al. 2015; Wang et al. 2014; Wang et al. 2015; Guo et al. 2009):

- Die Anwendung kann für eine konkrete Laufzeitumgebung entwickelt und optimiert werden. Dabei können Technologien optimal aufeinander abgestimmt werden.
- Die gesamte Infrastruktur kann für den Zugriff auf Hintergrundsysteme optimiert werden.

3.2.2 Client-Zentrierung

„Rein“ clientseitige HTML5-Anwendungen gibt es normalerweise nicht. Die Anwendung wird ebenfalls von einem Server ausgeliefert, im einfachsten Fall als statische JavaScript-Datei. Im Normalfall muss sie jedoch zusätzlich mit Daten versorgt werden. Daten können zwar grundsätzlich auch innerhalb der umgebenden HTML-Seite transportiert werden (Inline JavaScript-Code oder HTML-Elemente), jedoch vor allem durch die Bereitstellung von dynamisch generiertem JavaScript-Code (z.B. JSON), z.B. durch das Bereitstellen eines entsprechenden Dienstes und asynchrones Laden der Daten aus der JavaScript-Anwendung.

Die im Client (Browser) laufende Anwendung macht es nun möglich, direkt und ohne Umweg über den Server (Neuladen der Seite) auf Interaktionen und Ereignisse zu reagieren, insbesondere die Darstellung zu verändern (z.B. Sortierung in einer Tabelle) oder bei Bedarf weitere Inhalte nachzuladen.

Hierzu ist der Zugriff auf entsprechende Datendienste notwendig, die auf dem Heimatserver der Anwendung, jedoch auch auf davon unabhängigen Servern, bereitgestellt werden können.

Viele Probleme der rein serverseitigen Anwendungen können so gelöst werden:

- Flaschenhals Server-Anwendung
- Vermeidung des Nachladens von ganzen HTML-Seiten
- Hoch-interaktive und ergonomische Benutzeroberflächen.

Allerdings muss bei Browseranwendungen auch ein beträchtlicher Teil der Anwendungslogik im Client ausgeführt werden. Je nach Anwendungsfall werden daher bestimmte Anforderungen an die Leistungsfähigkeit des Clients und ggf. auch an die Bandbreite und Geschwindigkeit der Netzverbindung gestellt, da im Allgemeinen im Vergleich zur rein serverseitigen Lösung mehr einzelne Übertragungen notwendig sind und auch das zu übertragende Datenvolumen steigen kann. In der Praxis spielt das beispielsweise bei auf Mobilfunk basierenden Verbindungen eine Rolle.

Manche Teile einer Anwendung lassen sich ggf. nicht ohne Weiteres komplett auf den Client übertragen, insbesondere wenn damit der Zugriff auf bzw. die Verarbeitung von großen Datenmengen verbunden ist. Hier ist es meist sinnvoll, die betroffenen Teile der Anwendung in einen vom Client nutzbaren Service umzubauen, so dass die eigentliche Verarbeitung auf einem leistungsfähigen Server abläuft (auf dem ggf. auch die Daten gehalten werden und direkt zur Verfügung stehen), und im Client nur ein Stellvertreter (Proxy, Stub) des serverseitigen Dienstes vorhanden ist, der sich um den Aufruf des Dienstes und das Durchreichen der Ergebnisse kümmert. Der aus obigen Überlegungen resultierende hybride Ansatz wird im folgenden Abschnitt beschrieben.

3.2.3 Hybrider Ansatz

Mit der Ontologie-Komponente ist ein datenintensiver Teil vorhanden. Die Verarbeitung von Anfragen an das Ontologie-System sollte daher direkt auf dem Server ablaufen, auf dem auch das Ontologiesystem vorgehalten wird. Benötigt wird das System vor allem zum Erkennen der inhaltlich-thematischen Bedeutung einer Suchanfrage, und damit zur Präzisierung der Suchanfrage (Abbildung auf wohlbekanntes Wortgut), ggf. auch zur Anreicherung der Suchbegriffe um ergänzende Attribute.

Es bietet sich an, hierfür entsprechende Dienste zu implementieren, die zum Beispiel über eine REST-Schnittstellte aufgerufen werden können. Andere Teile des Search-Brokers lassen sich ebenfalls als solche Dienste realisieren, tatsächlich wurden Dienste zum Ermitteln des Ortsbezugs einer Suchanfrage auch im serverzentrierten Ansatz

bereits über externe Gazetteer-Dienste implementiert, die auch direkt in der Client-Seite genutzt werden könnten.

3.2.4 Nutzung von Web-Widgets

Bei Web-Widgets handelt es sich um kleine Softwarekomponenten (meist programmiert in JavaScript), die sich in eine HTML-Seite einbetten lassen. Jedes Web-Widget stellt dabei eine Mini-Anwendung mit einem spezialisierten Aufgabenbereich dar. Im Idealfall sind Web-Widgets hochgradig konfigurierbar. Einmal aufgerufen, agieren sie weitgehend autonom, d.h. sie laden (asynchron) die notwendigen Daten, kümmern sich um deren Darstellung und reagieren auf Nutzerinteraktionen. Grundsätzlich ist dabei eine Interaktion von Web-Widgets mit anderen Komponenten der HTML-Seite möglich. Ein klassisches Beispiel für ein Web-Widget ist eine Google-Maps-Karte, die sich durch Hinzuladen einer entsprechenden JavaScript-Bibliothek sowie das Hinterlegen einer „Konfiguration“ (in Form eines kleinen JavaScript-Programmes) in jede beliebige HTML-Seite einbetten lässt. Die Anzeige von Inhalten und das Reagieren auf Nutzerinteraktionen (z.B. Zoomen oder Verschieben des Kartenausschnitts) übernimmt dabei vollständig das Google-Maps-Widget.

Die Parametrisierung der Widgets erfolgt sowohl durch entsprechende Konfigurationsdateien als auch durch die clientseitige Kommunikation der Widgets untereinander, bei der beispielsweise Informationen zur Änderung des Ortsbezugs ausgetauscht werden. Jedes Widget kann individuell auf die Ereignisse reagieren und ggf. Daten nachladen oder die Darstellung ändern, ohne dafür die ganze Seite neu laden zu müssen.

Insgesamt stellt das Widget-Konzept eine flexible Art der Nutzung vorhandener Dienste dar, die insbesondere auch unabhängig vom verwendeten Basis-System ist. Richtig programmierte Web-Widgets lassen sich wiederverwenden und sowohl in statischen HTML-Seiten als auch in verschiedenen Content Management Systemen (CMS) (Baun et al. 2010; Christ 2003) und Portalsystemen (GlossarWiki 2017) einsetzen. Dennoch können sie durch die Einbettung in entsprechende systemabhängige Container in das Darstellungs- bzw. Konfigurationskonzept des umgebenden Systems integriert werden (Schlachter et al. 2014b).

3.2.5 Kopplung von Web-Widgets per Eventbus

Das Konzept kleiner unabhängiger Softwarekomponenten innerhalb einer HTML-Seite impliziert, wie oben beschrieben, die Notwendigkeit des Datenaustauschs zwischen Komponenten. Wird z.B. der Darstellungsbereich einer Kartenkomponente durch das Verschieben des Ausschnitts oder durch Zoomen verändert, sollen andere Komponenten, die Inhalte bezüglich des angezeigten Ortes filtern, auf die Veränderung hingewiesen werden und ihre Darstellung entsprechend anpassen. Die Ergänzung eines Begriffs im Suchschlitz soll das Neuladen von Suchergebnissen triggern, jedoch können auch andere Komponenten an den Suchbegriffen interessiert sein, z.B. sollen passen-

de Pegelwerte geladen und angezeigt werden, wenn dort der Name eines Fließgewässers eingegeben wurde.

Die Kopplung von Widgets lässt sich über die Verwendung von clientseitigen Ereignissen realisieren (Schlachter et al. 2014b). Global wird hierfür eine anwendungsspezifische Auswahl möglicher Ereignistypen definiert, z.B. das Senden eines Ortsbezugs als Latitude-Longitude-Paar oder als Bounding-Box.

Die Anwendungsinfrastruktur stellt einen Ereignis-Dienst (im Folgenden „Eventbus“) zur Verfügung, der beim Aufbau der Seite initialisiert wird. Alle enthaltenen Web-Widgets können sich beim Eventbus für einen oder mehrere Ereignistypen registrieren, Ereignisse empfangen und auch Ereignisse an den Eventbus senden. Der Eventbus stellt sicher, dass alle registrierten Widgets auf Veränderungen aufmerksam gemacht werden, die durch Nutzerinteraktionen oder durch andere Komponenten ausgelöst werden. Hierbei handelt es sich um eine lose Kopplung, d.h. die Widgets beeinflussen sich nicht direkt. Jedes Widget entscheidet selbst, ob und wie es auf Ereignisse reagiert. Die Widgets sind insofern gekapselt, was sich positiv auf deren Wiederverwendbarkeit auswirkt. Über die Differenzierung von Ereignissen lässt sich dennoch ein scheinbar enges Zusammenwirken der verschiedenen Komponenten (Widgets) erreichen.

Andere Ereignis-Systeme (z.B. die eines umgebenden Portal- oder CMS-Systems und derer Frontend-Komponenten) lassen sich an den Eventbus über Adapter oder Brücken (Bridge)-Module anbinden, so z.B. die Inter-Portlet-Kommunikation (IPC) des Liferay-Portal-Servers (Liferay 2016; Java Community Process 2008). Auch initiale Parameter (z.B. URL-Parameter aus dem Seitenaufruf oder serverseitig gespeicherte Personalisierungsdaten) lassen sich, ebenso wie clientseitige (z.B. aus Cookies oder dem Local Storage von HTML5-fähigen Browsern), über den Eventbus in die Anwendung ein- bzw. auskoppeln. (Schlachter et al. 2014b)

3.3 Erste Architekturvariante: Semantische Erweiterung von Suchanfragen und Nutzung externer Datenquellen durch die Volltextsuchmaschine

Die LUPO Landesumweltportale (Schlachter et al. 2008; Schlachter et al. 2014b) als fachliche Rechercheportale verwenden seit 2008 kommerzielle Suchmaschinentechnologie, hauptsächlich zur Indizierung von un- und semistrukturierten Daten, insbesondere von Websites und Dokumentenbeständen, aber auch von ausgewählten strukturierten Datenbeständen, z.B. aus relationalen Datenbanken. Die bis heute im Einsatz befindliche Suchmaschine Google Search Appliance (GSA) wurde unter anderem wegen der hohen Relevanz ihrer Suchergebnisse ausgewählt, jedoch bereits bei der Erstellung von Kriterien zur Auswahl einer Suchmaschine und bei den ersten Konzepten zu deren Einbindung in die Landesumweltportale spielten strategische Überlegungen bezüglich der semantischen Erweiterung von Suchanfragen sowie die Einbeziehung wei-

terer Informationssysteme (außerhalb des Volltextindex) eine zentrale Rolle (Schlachter et al. 2008). Die strategischen Überlegungen zielen auf das Verfügbarmachen von Umweltinformationen als Linked Data für das Semantic Web.

Die entsprechende Umsetzung ist in Abbildung 3 dargestellt. Die Suchmaschine Google Search Appliance (GSA) greift direkt auf die Zielsysteme ZS_i zu. Alle gelesenen Daten, dabei handelt es sich im Wesentlichen um unstrukturierte bzw. semistrukturierte Daten wie HTML-Seiten oder PDF-Dokumente, werden im Index der Suchmaschine abgelegt. Die Suche erfolgt im Kern textvergleichbasiert mit vielfältigen Relevanzkriterien und kann verschiedene Wörterbücher, z.B. der deutschen Sprache (Wortstämme), nutzen.

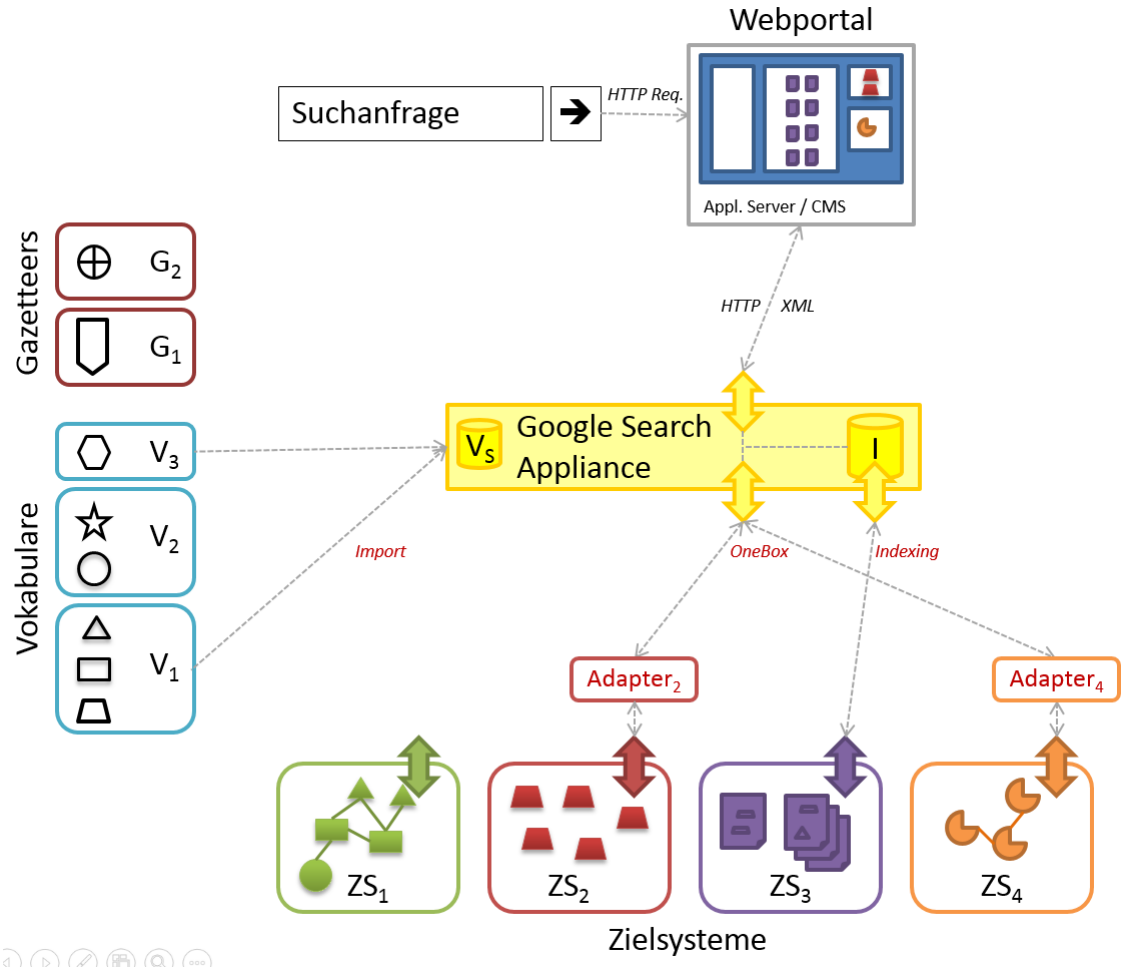

Abbildung 3: Umsetzung auf Basis einer vorhandenen Volltextsuchmaschine (neue Entwicklungen und eigene Anteile in rot)

Als mögliches gemeinsames Vokabular steht der Umweltthesaurus UMTHES des Umweltbundesamtes (Domäne „Umwelt“) zur Verfügung, der im Rahmen der Semantic Network Services (SNS) des Umweltbundesamtes (Umweltbundesamt 2016) elektronisch bereitgestellt wird. Das Vokabular kann die Suche ergänzen, indem Informationen aus dem Umweltthesaurus, z.B. Synonymketten (Benennung gleichartiger Objekte anhand verschiedener Suchbegriffe), in die Suche integriert werden.

Weitere, bisher nicht indexierte Datenquellen bzw. Zielsysteme (im Bild ZS_2 und ZS_4) lassen sich, falls notwendig über Adapter (im Bild $Adapter_2$ bzw. $Adapter_4$), über die OneBox-Schnittstelle der GSA parallel zur Index-basierten Suche abfragen und zusammen mit der klassischen Volltext-Trefferliste präsentieren.

3.3.1 Semantische Erweiterung von Suchanfragen

Die wesentliche Motivation für die semantische Erweiterung der Suchanfragen war die Beobachtung, dass die von Anwendern verwendeten Suchbegriffe häufig nicht der in Fachdokumenten gebräuchlichen Fachterminologie entsprachen und die auf dem Vergleich von Zeichenketten basierende Suchmaschine so häufig keine oder zu wenige Treffer lieferte. Durch die Integration eines Fachthesaurus (GEMET/Semantic Network Service (SNS)) (Bandholtz; Umweltbundesamt 2016; Angrick et al. 2002; Rüther und Bandholtz 2008), der in seinen Synonymketten auch umgangssprachliche Begriffe enthielt, als Suchmaschinen-Vokabular V_S konnte mit relativ geringem Aufwand eine deutliche Verbesserung erreicht werden, eine Suche nach „Müll“ enthielt beispielsweise auch Treffer zum behördlichen Begriff „Abfall“ und dem Unterbegriff „Schrott“.

Neben der Erweiterung der Suchanfragen durch Wörterbücher und Thesauri ist mit der GSA auch die manuelle Pflege von besonders relevanten Ergebnissen für bestimmte Suchanfragen (Key-Matches) (Google Inc. 2014a) möglich. So kann zum Beispiel bei der Eingabe eines Begriffes wie „Feinstaub“ auf eine passende Webseite hingewiesen werden.

Zusätzlich lassen sich in der Ergebnisliste Hinweise auf verwandte Suchbegriffe (Google Inc. 2014b) einblenden, z.B. um Benutzer beim Aufkommen aktueller Umweltfragen wie „Feinstaub“ oder „Gammelfleisch“ noch vor der Aufnahme in einen Fachthesaurus auf die entsprechenden Fachtermini (“PM10” bzw. “Lebensmittelhygiene”) hinzuweisen.

3.3.2 OneBoxes zur Einbindung externer Datenquellen

Die Notwendigkeit einer Möglichkeit zur Einbeziehung weiterer Informationssysteme (außerhalb des Suchindex) in eine Suchanfrage hat im Wesentlichen drei Gründe:

1. Nicht alle Quellsysteme lassen sich mit den üblichen Mechanismen (Web-Crawler, Verfügbare Konnektoren der Suchmaschine) indexieren
2. Eine Indizierung vieler Datentypen durch eine Suchmaschine ist nicht sinnvoll (z.B. häufig aktualisierte Messdaten)
3. Eine Indizierung ist aus lizenztechnischen bzw. Kostengründen (Anzahl der zulässigen Objekte/Dokumente im Suchindex zu teuer) nicht möglich.

Die GSA bietet eine flexible Möglichkeit zur Erweiterung der Ergebnisliste, sogenannte OneBoxes (Google Inc. 2015). Zum Beispiel ausgelöst durch die Verwendung bestimmter Suchbegriffe können neben der eigentlichen Ergebnisliste weitere Suchergebnisse eingeblendet werden. Solche Suchergebnisse können entweder aus einer

parallelen Suche auf einem Teilbereich des Suchindex kommen oder durch die Online-Abfrage weiterer externer Informationssysteme gewonnen werden.

Die externen Informationssysteme müssen hierzu die (proprietäre) OneBox-Schnittstelle implementieren, die als Eingabe im Wesentlichen die verwendeten Suchbegriffe in Form eines URL-Parameters bekommt. Für die Ausgabe und Datenübermittlung an die GSA ist ein (ebenfalls proprietärer) XML-Dialekt festgelegt, der für jeden Treffer die Angabe eines Titels sowie einer URL (als Sprungziel) verlangt. Darüber hinaus lässt das Format für jeden Treffer eine generische Liste weiterer Schlüssel-Wert-Paare zu.

Obwohl zum Zeitpunkt der Einführung der GSA für die Landesumweltportale keine entsprechenden OneBox-kompatiblen Dienste existierten, konnten dank des simplen Mechanismus innerhalb weniger Wochen eine ganze Reihe externer Datenquellen an die GSA angeschlossen und damit innerhalb der Landesumweltportale verfügbar gemacht werden. Dazu wurden meist kleine Proxy-Dienste als Adapter-Software implementiert, d.h. kleine Softwaremodule, die auf der einen Seite die OneBox-Schnittstelle implementierten und auf der anderen Seite an eine existierende Schnittstelle oder Datenquelle angebunden waren. Inhaltlicher Schwerpunkt waren aktuelle Messdaten wie Pegelwerte oder Luftqualitätsdaten sowie die Suchschnittstellen weiterer fachlich relevanter Portale (Statistische Landesamt, Verwaltungsportal service-bw).

Auch wenn die Simplizität des OneBox-Mechanismus auf der einen Seite zum schnellen Anschluss weiterer Datenquellen beigetragen hat, offenbarte er auf der anderen Seite sehr schnell Grenzen. Insbesondere die Beschränkung der “Eingabe” auf die bei der Suchanfrage verwendeten Suchbegriffe stellte eine deutliche Einschränkung dar:

- OneBoxes profitieren nicht von der semantischen Erweiterung der Suchanfrage durch Wörterbücher und Thesauri.
- OneBoxes müssen Daten letztlich auf Basis von einem oder wenigen Begriffen liefern, ihnen stehen keine weiteren Informationen und kein weiterer Kontext zur Verfügung.
- OneBoxes agieren autark, d.h. ohne Informationen darüber, ob parallel weitere Informationssysteme angefragt wurden bzw. welche Daten von dort geliefert wurden.

Konsequenz daraus ist, dass z.B. Pegel (zur Bestimmung des Wasserstandes in Gewässern) nur auf Basis des Namens (meist ein Ortsbegriff wie “Maxau”) oder des Gewässernamens (“Rhein”) gefunden werden. Wird ein davon abweichender Ortsbegriff wie “Karlsruhe” verwendet, der in den Metadaten keines Pegels enthalten ist, liefert der OneBox-Dienst keine Treffer. Um die Funktionalität einer Ortssuche mit Umkreissuche als OneBox realisieren zu können, muss der OneBox-Dienst die Explizierung der enthaltenen Ortsbegriffe sowie die Umkreissuche selbst implementieren, z.B. durch Nutzung eines Gazetteer-Services, vorausgesetzt die Metadaten der Pegel enthalten eine explizite Ortsinformation (Geokoordinaten). Hat ein OneBox-Dienst die Ortssuche tat-

sächlich realisiert, steht die gewonnene Information jedoch weiteren OneBoxen nicht zur Verfügung.

Ein weiterer Nachteil des OneBox-Mechanismus besteht in der Art und Weise wie OneBox-Dienste durch die GSA aufgerufen und ihre Ergebnisse verarbeitet werden. Der Anspruch, die Ergebnisse von OneBox-Diensten gemeinsam mit dem Suchergebnis auszuliefern, setzt der möglichen Verarbeitungsdauer inklusive Latenzen eine (konfigurierbare) Obergrenze. Die Qualitätsanforderungen an die Google-Volltextsuche zwingen alle angefragten OneBoxen innerhalb eines bestimmten Timeouts zu antworten. Liegt der GSA bis zum konfigurierten Zeitpunkt keine Antwort einer OneBox vor, so werden die Suchergebnisse ohne die möglichen Zusatzinformationen der betreffenden OneBox ausgeliefert und stehen somit in der Trefferansicht nicht zur Verfügung.

Umgekehrt wird aber auch die GSA-Suche durch lange laufende OneBoxes regelmäßig bis zum eingestellten Timeout (z.B. 1000 ms) ausgebremst, so dass sich die Suche für den Nutzer spürbar verlangsamen kann.

Das Ausliefern eines monolithischen Ergebnis-Dokuments (XML) mit allen Teilergebnissen führt dazu, dass zur Anzeige der Suchergebnisse eine Abbildung von Teilergebnissen auf die passenden Anzeigekomponenten notwendig ist, d.h. sie muss in irgendeiner Form konfiguriert oder programmiert werden, im Falle der WebGenesis-basierten Umweltportale die Aufgabe eines Dispatchers, der den Daten eine Reihe von Templates zuordnen konnte.

In den Webgenesis-basierten Landesumweltportalen (Schlachter et al. 2008) werden die Trefferansichten komplett serverseitig prozessiert. Durch die Verwendung von Templates wird bereits im Server fertiger HTML-Code erzeugt, der so an den Client (Webbrowser) übertragen wird, was letztlich zu starren und statisch wirkenden Anzeigen führt.

Eine direkte Auswirkung der serverseitigen Erzeugung von kompletten Trefferansichten ist, dass Interaktionen des Nutzers jeweils zu Änderungen der Suchanfrage führen, und damit jeweils ein Neuladen des gesamten Suchergebnisses (inklusive aller OneBoxen) nötig ist, d.h. Teilergebnisse können nicht unabhängig von anderen nachgeladen werden.

3.3.3 Bewertung

Auch wenn der OneBox-Mechanismus durchaus Schwächen hat, so hat er sich für einige Anwendungsfälle bewährt, z.B. für die Suche in Metadaten zu Medien oder die kaskadierende Suche in weiteren Suchmaschinen (Statistisches Landesamt, service-bw), und wurde im Bereich der LUPO Landesumweltportale jahrelang intensiv genutzt.

Es ergeben sich jedoch daraus einige direkte Folgerungen:

- Die „semantische Suche“ in der vorgestellten ersten Architekturvariante bezieht sich lediglich auf die Verwendung von Synonymketten für die Erweiterung der textvergleichbasierten Volltextsuche. Sie führt zu einer echten Verbesserung

der Suche, z.B. durch die Möglichkeit zur Verwendung umgangssprachlicher Begriffe. Eine tatsächliche semantische Verarbeitung der verwendeten Suchbegriffe sowie der Abbildung auf die Konzepte der Daten in den Zielsystemen findet jedoch nicht statt. Sie soll in die Suche integriert werden.

- (Semantische) Erweiterungen bzw. Explizierungen der Suchanfrage sollen auch bei der Anfrage aller möglichen Zielsysteme bzw. Datenquellen (hier: auch bei OneBox-Systemen) zur Verfügung stehen.
- Abfrageschnittstellen (APIs) der Zielsysteme ZS_i müssen den Anforderungen der jeweiligen Anwendungsfälle gerecht werden (z.B. Umkreissuche mit Mittelpunkt und Radius).
- Suchergebnisse bzw. Daten aus mehreren/verschiedenen Zielsystemen/Datenquellen sollen unabhängig voneinander (asynchron) angefragt werden können.
- Zielsysteme sollen einheitliche Schnittstellen und Datenformate verwenden, um die Möglichkeit zur Wiederverwendung von (generischen) Komponenten zu verbessern und somit den Aufwand für deren Entwicklung zu reduzieren.
- Die Bereitstellung von Daten in separaten Diensten (Services) ist sinnvoll, insbesondere wenn sie nicht über eine Suchmaschine indexiert werden können oder wenn die Bereitstellung in einem separaten Dienst über eine spezifische Abfrageschnittstelle (API) bzw. in einem spezifischen Format einen Mehrwehrt für die Anwendung bietet.
- Der “Zwang” zur Nutzung von spezifischen (ggf. nicht standardisierten oder proprietären) Schnittstellen führt dazu, dass existierende Systeme, welche die Schnittstellen nicht zur Verfügung stellen können, über Adapter/Proxy angebunden werden müssen.

Weitere Bemerkungen und Erkenntnisse

- In der OneBox-Architektur tritt explizite Semantik bestenfalls bei der Erweiterung von Suchanfragen (durch Thesauri und Wörterbücher) zutage. Sie steckt jedoch auch hier implizit in den verwendeten Thesauri und kommt bei den Anfragen an die OneBoxes nicht zur Anwendung.
- “Wahlloses” Anfragen von Diensten (hier: OneBoxes) mit allen beliebigen Suchbegriffen führt zu einer Vielzahl von erfolglosen Anfragen, d.h. Anfragen ohne Ergebnis.

Wenn die obigen Verbesserungen in die semantische Suche in heterogenen Informationssystemen eingebracht werden sollen, ist eine Modifikation der hier vorgestellten Architekturvariante erforderlich. Damit beschäftigen sich die folgenden Abschnitte.

3.4 Zweite Architekturvariante: Serverseitige Verarbeitung der Suchanfrage, SearchBroker und Ontologiesystem

Den Kern der zweiten Architektur-Alternative (Abbildung 4) bildet ein Portal, das möglichst alle verfügbaren Informationen einer Domäne für den Nutzer (Fachnutzer bzw.

die allgemeine Öffentlichkeit) zugänglich machen soll. Die Informationen liegen dabei nicht im Portal selbst, sondern in dedizierten Systemen, die lediglich über Schnittstellen in das Portal integriert und dort recherchierbar gemacht werden.

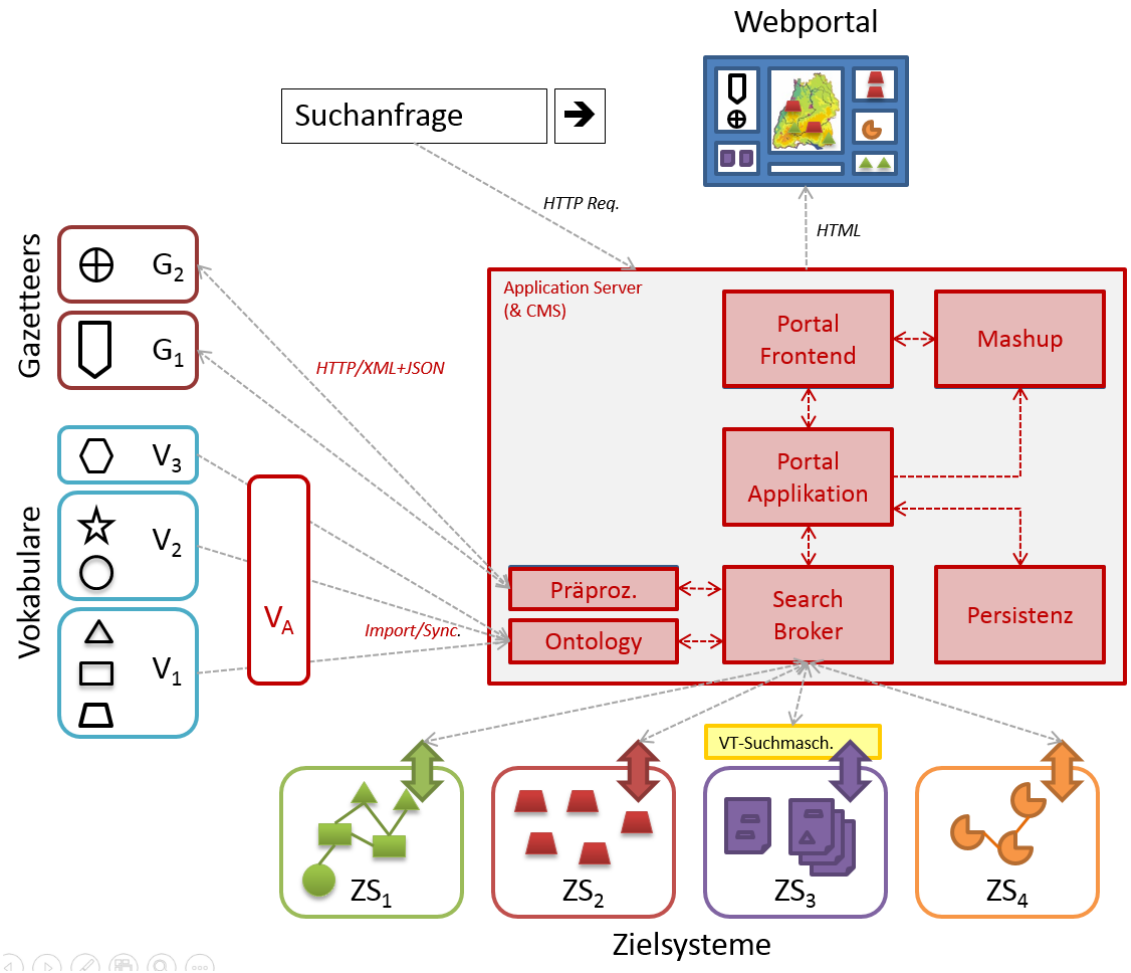

Abbildung 4: Serverseitige Umsetzung mit SearchBroker und Ontologiesystem (neue Entwicklungen und eigene Anteile in rot)

Das (Such-)Portal selbst besteht aus verschiedenen Komponenten, die dabei jeweils weitgehend unabhängig von anderen Komponenten, spezialisierte Aufgaben (z.B. Vorverarbeitung der Suchanfrage, Abfrage von Zielsystemen, die Visualisierung bestimmter Datentypen etc.) ausführen.

Der modulare Ansatz bietet weitreichende Vorteile, insbesondere die Wiederverwendbarkeit von Modulen, die Möglichkeit zum Zusammenstellen von Anwendungen als Kombination von Modulen (Baukasten) und die Möglichkeit zum Austausch von Modulen gegen neue, bessere, leistungsfähigere oder alternative Module.

Alle Komponenten zur Verarbeitung der Suchanfrage sind dabei in ein Portal auf Basis von WebGenesis, einer Software zur Generierung von Web-basierten Informationssystemen (Fraunhofer IOSB 2016), implementiert. Abbildung 5 zeigt eine Grobübersicht der Komponenten des Portals.

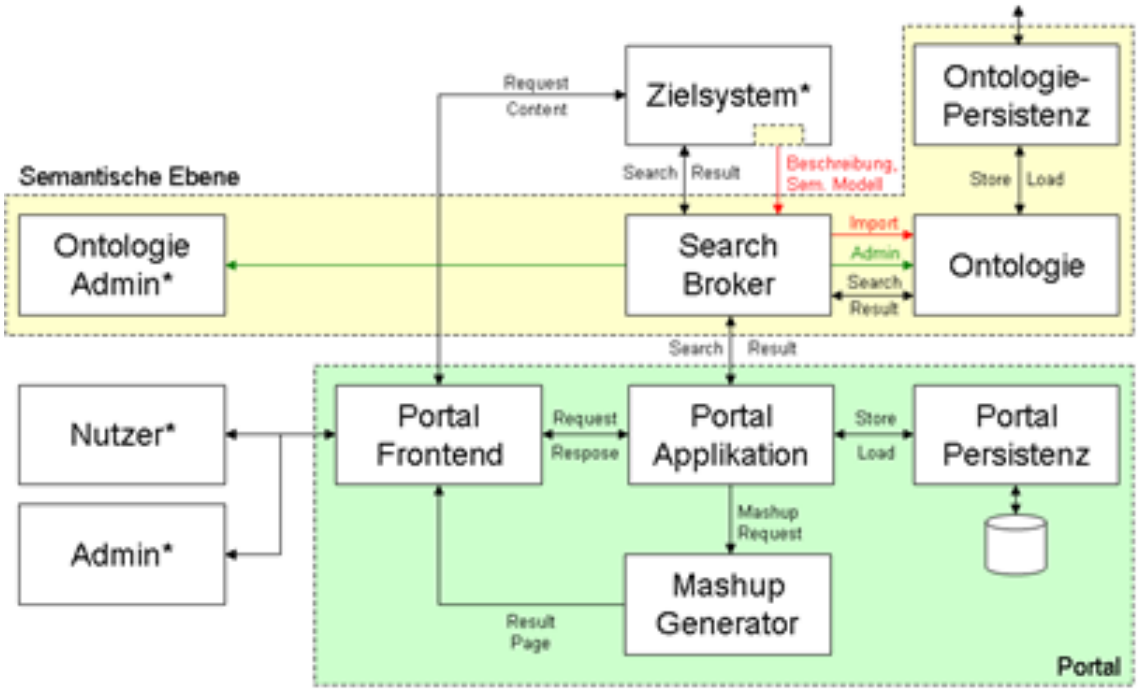

Abbildung 5: Übersicht über die Komponenten des Portals

Im unteren, grünen Teil befinden sich dabei die Komponenten des Portals selbst, der obere, gelbe Teil beinhaltet die Komponenten der semantischen Ebene (die semantische Suche).

Im Portal sind die Generierung der eigentlichen Nutzeroberfläche (Portal Frontend), die Applikationslogik (Portal Applikation), eine Daten-Persistenzschicht zur Speicherung von (Meta-)Daten (Portal Persistenz) innerhalb des Portals sowie eine spezialisierte Komponente zur Aufbereitung der Ergebnisansichten für Suchanfragen (Mashup Generator) enthalten. Nutzer und Administratoren kommunizieren dabei ausschließlich mit dem Portal Frontend. Das Frontend kann prinzipiell (modularer Ansatz) gegen andere Frontend-Komponenten ausgetauscht werden, die zum Beispiel alternative Ansichten, etwa für die Nutzung der Anwendung auf mobilen Geräten, implementieren können.

Gegenüber dem Portal tritt die semantische Suche als eine mögliche, ggf. austauschbare Suchmaschine auf. Die Anbindung an das Portal bildet die Abfrageschnittstelle des SearchBrokers. Die definierte Schnittstelle soll eine Austauschbarkeit der semantischen Suche gegen andere Suchverfahren (z.B. konventionelle Volltextsuchmaschinen) sicherstellen, insbesondere in umgekehrter Weise, bei der konventionelle Suchmaschinen gegen eine semantische Suche ausgetauscht werden.

3.4.1 SearchBroker als zentrale Komponente der Suche

Der SearchBroker stellt die Kernkomponente der semantischen Ebene dar (Abbildung 6).

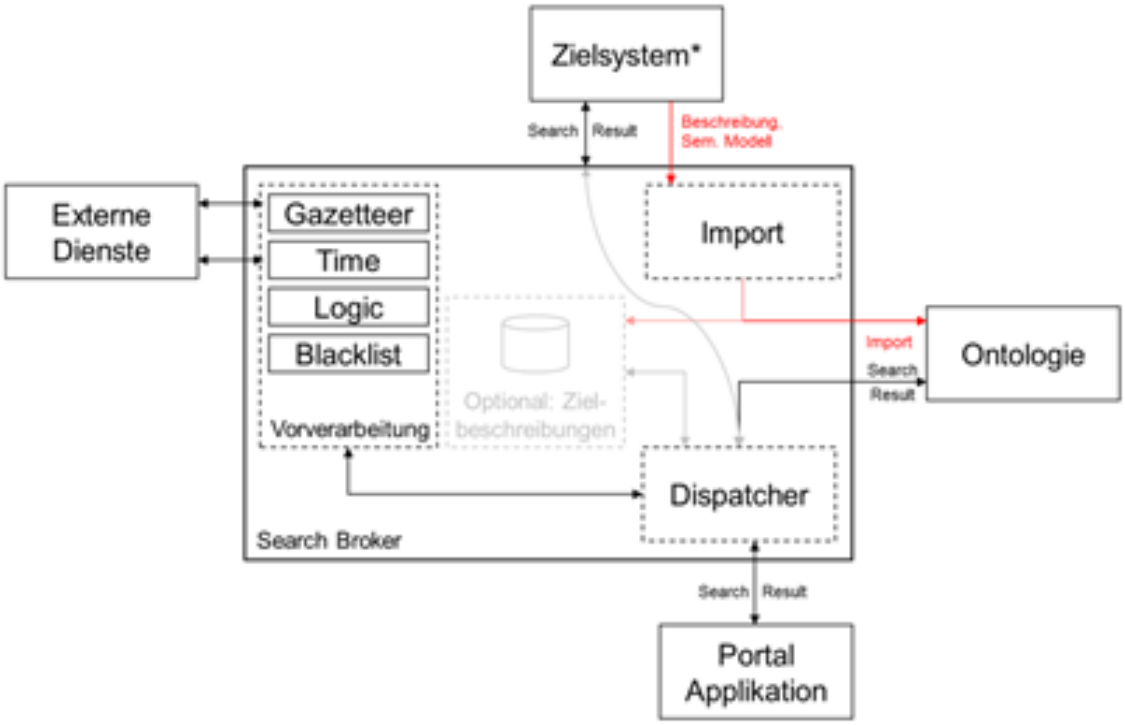

Abbildung 6: Übersicht über den SearchBroker

Er verfügt über Informationen zu allen angeschlossenen Zielsystemen in Form von Zielsystembeschreibungen. Jede Zielsystembeschreibung enthält einerseits Informationen zum Zugriff auf das jeweilige Zielsystem (technische Schnittstellen), andererseits aber auch Informationen zur Art und Bedeutung (Semantik) der im Zielsystem abrufbaren Informationen. Die Verfügbarkeit der Semantik hebt den SearchBroker von konventionellen Volltextsuchmaschinen ab, die meist nicht über Metainformationen zur Semantik der indexierten Daten verfügen und die Einordnung (Gewichtung) von Inhalten z.B. aus deren Struktur (etwa Auszeichnung in Form von HTML-Tags) vornehmen müssen. Wenn die Interpretation einer HTML-Tabelle, die in jeder Zeile aktuelle Ozon-Messwerte zu einer Messstation enthält, nur auf einer strukturellen Ebene (in der ersten Spalte steht eine Zeichenkette, in der zweiten Spalte steht eine Zahl), jedoch nicht auf einer semantischen Ebene (in der zweiten Spalte steht der aktuelle Ozon-Messwert in µg/m^3 zu dem in der ersten Spalte genannten Ort) geschieht, gehen die Möglichkeit zur Nutzung der Informationen innerhalb des Suchergebnisses weitgehend verloren.

Der SearchBroker setzt sich aus verschiedenen Komponenten zusammen. Der Dispatcher ist für die Entgegennahme und Analyse von Suchanfragen zuständig. Im Wesentlichen untersucht er die Suchanfrage auf ihre semantischen Bestandteile (Elemente). Eine Anfrage wie „ozonmesswerte in karlsruhe im sommer 2013“ kann dabei z.B. rein syntaktisch (durch Trennung an den Leerräumen) in die Elemente „ozonmesswerte“, „in“, „karlsruhe“, „im“, „sommer“ und „2013“ aufgeteilt werden. Der menschliche Nutzer sieht auf den ersten Blick, dass die Anfrage semantisch aus drei Elementen besteht: Dem Thema „ozonmesswerte“, der Ortsangabe „in karlsruhe“ und dem Zeitraum „im sommer 2013“. Der Dispatcher hat die Aufgabe, solche semantischen Elemente zu

erkennen, um sie bei der Anfrage von Zielsystemen spezifisch einsetzen zu können. Dabei ist es sinnvoll oder sogar notwendig, die erkannten Elemente weiter zu explizieren, z.B. zu erkennen, dass es sich bei „in karlsruhe“ um die Stadt „Karlsruhe“ handeln könnte, oder das „im sommer 2013“ den Zeitraum “vom 21.06.2013 bis zum 22.09.2013“ bedeuten kann. Solche Interpretationen müssen nicht eindeutig sein, könnte es sich bei „in karlsruhe“ doch um die Stadt, den Regierungsbezirk, den Landkreis oder einen anderen Ort mit dem Namen “Karlsruhe” handeln, etwa einen Teilort der Gemeinde Plattenburg in Brandenburg oder ein Dorf in North Dakota (Vereinigte Staaten von Amerika).

3.4.2 Spezialisierte Plugins zur semantischen Vorverarbeitung der Suchanfrage

Um dem SearchBroker bei der semantischen Auflösung der Elemente in der Suchanfrage zu helfen, stehen ihm zur (Vor-)Verarbeitung der Suchanfrage eine Reihe von spezialisierten Komponenten (Plugins) zur Verfügung, die sich jeweils um die Erkennung von bestimmten Elementen der Suchanfrage kümmern und andere, unbekannte Elemente ignorieren. So ist das Plugin „Gazetteer“ z.B. auf die Erkennung von deutschen Ortsnamen spezialisiert, kann dabei auch räumliche Gliederungen wie Regierungsbezirke oder Landkreis verarbeiten und deren Beziehungen zueinander (Stadt Karlsruhe liegt im Regierungsbezirk Karlsruhe, liegt im Land Baden-Württemberg) auflösen. Eine andere Komponente („Time“) kann z.B. auf die Erkennung von zeitlichen Begriffen spezialisiert sein, z.B. aus „im sommer 2013“ die entsprechenden Anfangs- und Enddaten generieren, ggf. auch alternative Interpretationen (meteorologischer Sommer, astronomischer Sommer) liefern.

Für die Erkennung bestimmter Elementtypen kann es auch mehrere solcher Plugins geben, die, je nach Konfiguration parallel oder kaskadierend, Interpretation mit weiteren Daten anreichern. So kann z.B. das erste Gazetteer-Plugin aus „in karlsruhe“ die „Stadt Karlsruhe“ und den „Regierungsbezirk Karlsruhe“ erkennen, und ein weiterer spezialisierter Gazetteer-Dienst daraus anschließend den amtlichen Gemeindeschlüssel („Gemeindekennziffer“) „08 2 12 000“ zur „Stadt Karlsruhe“ bestimmen.

Die Ermittlung von speziellen Elementen ist auf Basis von vorgegebenen Wertelisten (Liste aller Ortsnamen Deutschlands) oder auf Basis von syntaktischen Mustern/Regeln wie (hier für die Jahreszeiten eines bestimmten Jahres)

```
(frühling|sommer|herbst|winter)\s+\d{4}
```

relativ einfach.

3.4.3 Auflösung thematischer Bezügen durch die Nutzung von Ontologien

Schwieriger ist dagegen das Erkennen eines Themenbezugs, z.B. aus dem Begriff „ozonmesswert“ einen inhaltlichen Bezug zu den Begriffen „Ozon“ und „Messwert“ und den dahinter stehenden Konzepten herzustellen. Selbst wenn man Teilbegriffe erkennen kann (z.B. durch Nutzung eines Wörterbuches auf die Zusammensetzung des Be-

griffes „ozonmesswert“ aus den Teilworten „Ozon“ und „Messwert“ schließen), heißt das nicht, dass auch die Semantik hinter den Teilbegriffen erfasst wurde. Es kann zum Beispiel auch relevante Synonyme, Unter- und Oberbegriffe geben oder eine Abbildung von umgangssprachlichen Begriffen auf die entsprechenden Fachbegriffe notwendig sein. So könnten in einem Fachsystem Ozonmesswerte z.B. nur unter der Bezeichnung „O_3“ bekannt sein.

Für die Erkennung/Auflösung des Themenbezugs bedarf es daher einer mächtigen Komponente, die über das notwendige (Fach-)Wissen innerhalb der Anwendungsdomäne(n) (z.B. „Umwelt“ und/oder „Energie“) verfügt. Die Beschränkung auf spezifische Domänen stellt eine deutliche Einschränkung dar, die jedoch auch mit einer erheblichen Vereinfachung einhergeht, da das „Wissen“ einer Domäne eine beschränkte Untermenge des „gesamten“ Wissens bildet. Innerhalb einer Domäne sind Fachbegriffe im Allgemeinen (genau/eindeutig) definiert, beispielsweise gibt es innerhalb einer Domäne keine/oder wenige Homonyme und Beziehungen zwischen Ober- und Unterbegriffen sind eindeutig bestimmt.

Innerhalb einer Domäne bilden die verwendeten Begriffe daher meist ein kontrolliertes Vokabular und es existieren in vielen Fällen Fachthesauri, Taxonomien oder andere verfügbare Systematiken, die sich in ihrer Darstellung, Komplexität und ihrer semantischen Reichhaltigkeit unterscheiden, s. Abbildung 7.

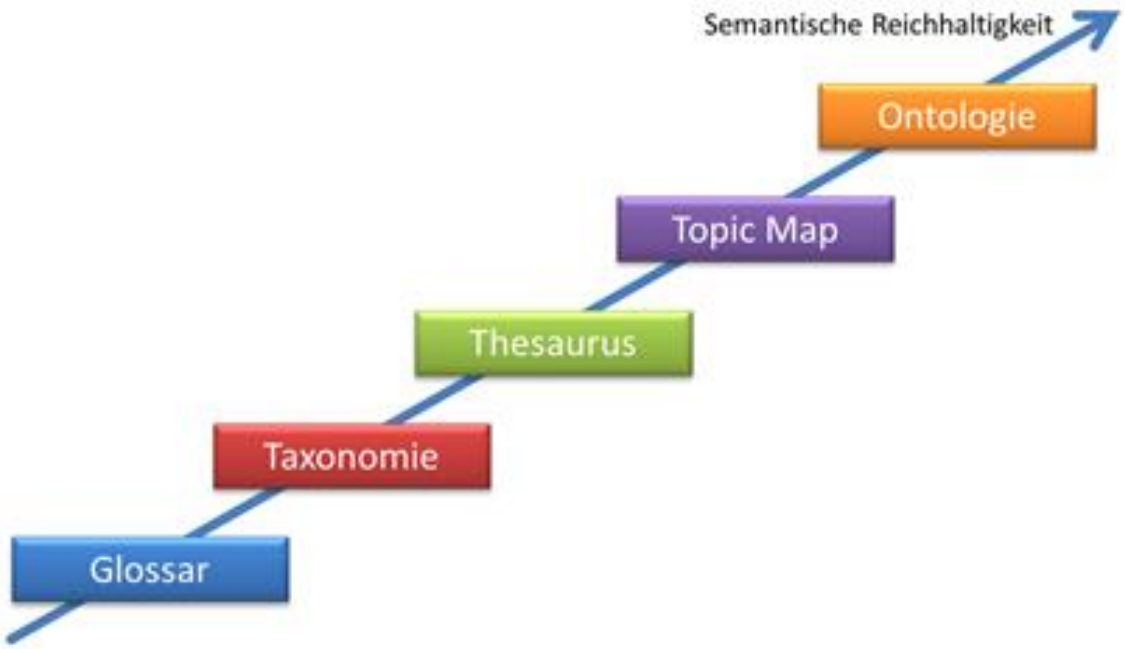

Abbildung 7: Semantische Treppe nach (Pellegrini und Blumauer 2006)

Die semantische Reichhaltigkeit und strukturelle Flexibilität steigt in der Abbildung von links unten nach rechts oben an. Bei einem Glossar handelt es sich um eine einfache Liste von Begriffen mit deren Erläuterung. Beziehungen zwischen Begriffen werden in einem Glossar nicht berücksichtigt.

Taxonomien erweitern das Glossar um eine hierarchische Struktur, d.h. die Beziehungen zwischen Begriffen sind in Form einer Ordnung (Oberbegriff, Unterbegriff) dargestellt. Weitergehende Beziehungen zwischen Begriffen lassen sich jedoch in einer Taxonomie nicht darstellen.
In einem Thesaurus können Beziehungen zwischen Begriffen/Objekten durch fest definierte Relationen, z.B. Synonym, Oberbegriff, Assoziation etc., ausgedrückt werden. Durch die Verwendung unterschiedlicher Relationen können verschiedene Arten von Beziehungen ausgedrückt werden, die statische Hierarchie einer Taxonomie wird aufgelöst.
Topic Maps sind als technische Standards (TopicMaps.Org 2001; ISO/IEC 2006) definiert und werden technisch meist auf Basis von XML-Dokumenten ausgedrückt. Auch bei Topic Maps werden Beziehungen zwischen Objekten ausgedrückt, sie lassen sich im Gegensatz zu den Beziehungen innerhalb eines Thesaurus jedoch frei definieren.
In einer Ontologie werden in der Informationsverarbeitung formelle Beschreibungen der Beziehungen zwischen Objekten definiert. Objekte werden konzeptionalisiert, d.h. Konzepten/Klassen zugeordnet, welche die Gemeinsamkeiten von Objekten zum Ausdruck bringen. Hinzu kommt eine frei definierbare Menge möglicher Beziehungen zwischen Konzepten, inklusive frei wählbarer Attribute zur Beschreibung einer konkreten Beziehung (Berners-Lee et al. 2001). Beziehungen lassen sich in einer Ontologie als Tripel Subjekt-Prädikat-Objekt darstellen, technische Formate wie RDF (W3C 2004c) verwenden solche Tripel. Ganze Ontologien können z.B. in Form von OWL-Dokumenten (W3C 2004a, 2012a) dargestellt werden.

Ist das Wortgut (Vokabular) einer Domäne eindeutig beschrieben, z.B. in Form einer Taxonomie, eines Thesaurus, einer Topic Map oder einer Ontologie, so lässt sich der thematischen Bezug einer Suchanfrage innerhalb der Domäne mit Hilfe des Wortguts im Allgemeinen präziser beantworten als allgemeine Fragen, wie sie z.B. von Internet-Suchmaschinen beantwortet werden.

Die Vokabulare von Domänen (Umwelt) liegen häufig in Form von Thesauri vor. Dort liegen im Allgemeinen eindeutige Benennungen für jeden einzelnen Begriff vor, die sog. „Deskriptoren“. Unterschiedliche Schreibweisen eines Begriffes, Synonyme, Abkürzungen und Übersetzungen werden in einem Thesaurus per Relation dem jeweiligen Deskriptor zugeordnet. Die Abbildung unterschiedlicher Begriffe auf denselben Deskriptor beschreibt bereits eine Beziehung, z.B. deren Äquivalenz. Neben Synonymen kann ein Thesaurus auch weitere Beziehungstypen zwischen Begriffen enthalten, insbesondere Ober- und Unterbegriffe sowie verwandte Begriffe, die aber nicht dasselbe bedeuten, d.h. nicht denselben gemeinsamen Deskriptor haben. Daneben können auch Spitzenbegriffe („Top terms“ oder „Einstiegspunkte“) als solche markiert sein.

Deskriptoren eignen sich sowohl für die Beschreibung von thematischen Bezügen in Suchanfragen als auch in Zielsystemen. Alle anderen Begriffe aus der Domäne lassen sich auf die zugehörige Deskriptoren abbilden.

3.4.4 Ontologiesystem

Zur Darstellung der verschiedenen Aspekte von Informationsquellen (thematische, örtliche Zuordnung, Lebenslage) werden auf inhaltlicher Ebene verschiedene, miteinander vernetzte Ontologien verwendet:

- **Domänen-Ontologie/Themen-Ontologie**: Auf Basis eines vorhandenen Vokabulars (z.B. einer Taxonomie, eines Thesaurus oder einer vollständigen Ontologie) modelliert die Domänen-Ontologie die thematischen Konzepte und Zusammenhänge in der spezifischen Domäne, z.B. „Umwelt“ oder “Energie” eines Informationssystems. Ggf. kann es sich auch um mehrere Domänen-Ontologien handeln, die miteinander zu vernetzen sind (s.u. „Artikulationsontologie“).
- **Lebenslagen-Ontologie**: In vielen Fällen kann der Kontext einer Anfrage helfen, deren Bedeutung weiter zu spezifizieren. Im Falle behördlicher Informationssysteme hat sich der Begriff der “Lebenslage” eingebürgert (Service-BW 2016). Die Möglichkeit zur Angabe einer Lebenslage kann dem Nutzer helfen, passende (z.B. vorgegebene oder erprobte) Muster für sein Anliegen zu finden, z.B. den Bau eines Wohngebäudes („Hausbau“). Lebenslagen müssen daher mit der Domänen-Ontologie vernetzt werden (s.u.), beispielsweise im Falle von “Hausbau” mit Energiethemen wie “Heizung”, “Dämmung”, “Regenerative Energien” etc.
- **Ortsontologie**: Grundsätzlich können auch Geolokationen in einer Teilontologie gespeichert werden. Es wird zwischen dem Repräsentationsformat der Eingabe, z.B. Ortsnamen, und denen der existierenden Datenbestände (Namen oder IDs administrativer Einheiten, Positionen) vermittelt. Die Ortsontologie kann entsprechend der Suchanfrage passende Geolokationen liefern. Im Falle des implementierten SUI-Portals (Abecker et al. 2009a; Bügel et al. 2010; Düpmeier 2012) steht zur Auflösung von Ortsbegriffen jedoch ein separater Gazetteer-Dienst als Plugin des SearchBrokers zur Verfügung, der Ortsnamen bereits im Vorverarbeitungsschritt expliziert. Sind Geolokationen in der Gesamtontologie integriert, müssen sie kompatibel zu den durch den Gazetteer ermittelten Lokationen sein. Sie erlauben dann eine Weiterverarbeitung der ermittelten Geolokationen, z.B. die Auflösung von Teilortbeziehungen oder das Ermitteln von benachbarten Orten (Abecker et al. 2009b).
- **Informationsontologie** (optional): Informationen über Informationsquellen (Zielsysteme) können ebenfalls in einer Ontologie abgelegt werden (Zielsystembeschreibungen) und z.B. thematische Zuordnungen (Bezug zur Domänen-Ontologie) sowie Informationen zum Zugriff auf das Zielsystem enthalten. Auf deren Basis kann die Anfrage eines Nutzers in Anfragen an die angeschlossenen Zielsysteme transformiert werden.
- **Artikulationsontologie**: Die oben aufgeführten Ontologien stellen noch disjunkte Teilontologien dar, d.h. es gibt keine Bezüge zwischen den Konzepten und Instanzen der Teilontologien. Zur Verbindung der Teilontologien wird daher eine weitere Ontologie, eine sogenannte Artikulationsontologie (Nikolai 2002)

hinzugefügt, die quasi den Klebstoff zwischen den einzelnen Teilen darstellt, sie kann darüber hinaus auch innerhalb einer Teilontologie Beziehungen darstellen, falls sie noch nicht enthalten sind, z.B. Relationen zwischen Ober- und Unterbegriffen.

Die Artikulationsontologie stellt zunächst Brücken zwischen identischen Konzepten der verschiedenen Ontologien in Form von Relationen her (Äquivalenz). Darüber hinaus können frei definierbare Relationen Verbindungen zwischen den Teilontologien herstellen, z.B. hat eine bestimmte Lebenslage wie „Hausbau“ mit Konzepten aus der Themen/Domänen-Ontologie zu tun, etwa „Heizung“, „Dämmung“ oder „Regenerative Energien“.

Jede Relation der Artikulationsontologie kann mit einem „Gewicht“, d.h. einem Attribut, das eine Wichtung der Relation repräsentiert, versehen werden. Die Gewichte können bei der Abfrage der Ontologien herangezogen werden, um die Umgebung gefundener Konzepte und Instanzen zu bestimmen. Äquivalenzrelationen können z.B. das Gewicht 0 erhalten, so dass im Ergebnis der Abfrage sowohl Subjekt als auch Objekt der Äquivalenzrelation enthalten sind, sobald entweder das Subjekt oder das Objekt einen direkten Treffer darstellt. Ober- bzw. Unterbegriffbeziehungen kann ein höheres Gewicht, z.B. 1, zugewiesen werden. Die abfragende Anwendung kann so ein maximales Gewicht angeben, bis zu welchem die Umgebung von direkten Treffern mitgeliefert werden soll. Resultat ist damit ein zusammenhängender Ausschnitt der Gesamtontologie, potenziell mit Bestandteilen aus verschiedenen Teilontologien (Nikolai 2002; Bügel et al. 2011c).

Da die Ontologieentwicklung als evolutionärer Prozess verstanden werden muss, sieht die Architektur des Ontologiesystems die Nutzung eines integrierten Ontologiemanagements vor. Neben der persistenten Speicherung der Ontologie, z.B. in einer relationalen Datenbank, umfasst das Ontologiesystem Funktionen für die Weiterentwicklung, das Mapping unabhängig voneinander entstandener Ontologieteile, die Anbindung von Inferenzmaschinen und Visualisierungskomponenten, die Population von Instanzdaten, die Entwicklung von Abfragen sowie die Verwaltung von Nutzer- und Provenance-Daten (d.h. Angaben zum Datenursprung).

Um für künftige Erweiterungen gerüstet zu sein, kommt den Schnittstellen des Ontologiesystems besondere Bedeutung zu. Deshalb wird als Schnittstelle die OWL-API (Sourceforge.net 2016) eingesetzt, eine Anwendungsinfrastruktur für die Integration von Management-Diensten. Die OWL-API erlaubt die einfache Manipulation von Ontologien nach dem OWL-Standard auf hoher Abstraktionsebene. Mit Hilfe von weiteren Adaptern können jedoch auch einfachere Ontologien nach dem RDF(S)-Standard über dieselbe Schnittstelle genutzt werden (Abecker et al. 2009b).

Suchanfragen können mit Hilfe des Ontologiesystems semantisch erfasst, d.h. einem oder mehreren Deskriptoren aus den Bereichen der Domäne, ggf. einer oder mehreren Lebenslagen, Orten und weiteren per Ontologie verknüpften Inhalten zugeordnet wer-

den. Die so gewonnenen Informationen können dann für möglichst konkrete Anfragen an die angeschlossenen Zielsysteme verwendet werden.

3.4.5 Zielsysteme und Zielsystembeschreibungen

Alle zu durchsuchenden Zielsysteme müssen der Suche bekannt gemacht werden. Einheitliche Zielsystembeschreibungen enthalten jeweils die notwendigen Metainformationen zu jedem Zielsystem. Kern jeder Zielsystembeschreibung ist eine technische und inhaltliche Beschreibung der Schnittstelle(n) zum Zugriff auf das Zielsystem.

Jedes Zielsystem wird in der Zielsystembeschreibung benannt (Name) und seine Schnittstelle(n) und die möglichen Rückgabeformate beschrieben. Der Name und ggf. eine eindeutige zugeordnete ID gewährleisten die Unterscheidung von Zielsystemen, wenn z.B. unterschiedliche Daten aus derselben Datenquelle abgerufen werden sollen.

Die Schnittstellenbeschreibung liegt in Form einer OpenSearch-Description (Google Inc. 2015; opensearch.org 2013) vor. Schnittstellen werden als URL-Muster mit Platzhaltern für Parameter beschrieben. Die Parameter sind durch Inhalte zu füllen, die aus der Suchanfrage selbst sowie aus der Vorverarbeitung der Suchanfrage gewonnen werde. Dazu müssen die Parameter syntaktisch und semantisch beschrieben werden, z.B. in welcher Form Geokoordinaten übergeben werden, etwa als Paar von Latitude (Breite) und Longitude (Länge), die jeweils in Dezimalgrad anzugeben sind, oder in Form einer Location-ID (z.B. eines Gemeindeschlüssels), falls Geodaten im Zielsystem vorhanden sind und so referenziert werden können. OpenSearch-Description bieten die Möglichkeit mehrere unterstützte Rückgabeformate aufzuzählen und für jedes mögliche Rückgabeformat eine eigene Schnittstellenbeschreibung anzugeben. Das lässt dem konsumierenden System (hier: dem Search Broker) die Wahl, in welcher Form die gewünschten Daten abgerufen werden sollen, z.B. wenn es mehrere mögliche Visualisierungsformen (z.B. Karte, Liste, Diagramm, Tabelle) für die Daten gibt.

Über weitere Metadatenfelder innerhalb der OpenSearch-Description können z.B. Kurzbeschreibung, Organisation etc. angegeben werden.

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/" xmlns:uis="https://www.lubw.de/uis/1.0/">
    <ShortName>UDO</ShortName>
    <Description>Definition von Suchmoglichkeiten in UDO</Description>
    <Tags>Schutzgebiet Landschaftsschutzgebiet Naturschutzgebiet Biotop Bannwald Schonwald</Tags>
    <Contact>admin@lubw.baden-wuertteMberg.de</Contact>
    <Url type="text/atom+xml"
         template="http://brsweb.lubw.de/cw?repoId=${uis:concept}_from_${uis:locType}&amp;val=${uis:locId}"/>
    <Url type="text/html"
         template="http://brsweb.lubw.de/cw?repoId=${uis:concept}_from_${uis:locType)&amp;val=${uis:locId}"/>
    <LongName>UMweltdaten and Karten Online Suche nach Schutzgebieten</LongName>
    <Image height="64" width="64" type="image/png">http://www.lubw.de/websearch.png</Image>
    <Image height="16" width="16" type="image/vnd.microsoft.icon">http://www.lubw.de/websearch.ico</Image>
    <Query role="example" uis:concept="naturschutzgebiet" uis:locType="gemeinde" uis:locationId="1234567"/>
    <Query role="example" uis:concept="landschaftsschutzgebiet" uis:locType="kreis" uis:locationId="80086"/>
    <Developer>disy</Developer>
    <Attribution>Search data Copyright 2010, www.lubw.de, All Rights Reserved</Attribution>
    <SyndicationRight>open</SyndicationRight>
    <AdultContent>false</AdultContent>
    <Language>de</Language>
    <OutputEncoding>UTF-8</OutputEncoding>
    <InputEncoding>UTF-8</InputEncoding>
</OpenSearchDescription>
```

Abbildung 8: Beispiel einer OpenSearch-Description (XML)

Abbildung 8 zeigt das Beispiel einer OpenSearch-Description im XML-Format. Die angegebenen URLs für den Aufruf enthalten Platzhalter der Gestalt `${searchTerms}`, `${uis:concept}`, `${uis:locType}` und `${geo:uid}`, über die Suchparameterwerte mit fest definierter Semantik beim URL-Aufruf an das Zielsystem übergeben werden können. Die Namen der eigentlichen Parameter innerhalb des URL-Aufrufs des Zielsystems sind dabei beliebig. Standardplatzhalter der OpenSearch-Spezifikation, wie z.B. „`searchTerms`“ zur Angabe von Volltextsuchbegriffen können ohne Präfix und Angabe eines Namensraums verwendet werden. Die im obigen Beispiel gezeigten Platzhalter mit Präfix „`uis`“ sind dagegen ergänzende Parameter und müssen daher gemäß der OpenSearch-Spezifikation durch einen eigenen, frei definierbaren Namensraum gekennzeichnet sein. Der Platzhalter „`geo:uid`“ stammt aus einer OpenSearch-Erweiterung, die von der OGC im Rahmen des Katalog-Standards definiert wurde, und beschreibt Suchparameter für eine räumliche Suche in Geodatensystemen.

Die Wahl des OpenSearch-Description-Formates schränkt die Auswahl möglicher Schnittstellen auf solche ein, denen eine URL (für einen GET-Request) zugrunde liegt. Das stellt zwar eine technische Einschränkung dar, nicht jedoch der Mächtigkeit der Schnittstelle, da weitere mögliche Schnittstellen, z.B. zum direkten Abfragen einer Datenbank per SQL, sich leicht über Adapter auf Basis von REST-Diensten, die eben über URLs adressierbar sind, realisieren lassen. Es fällt hier jeweils nur der zusätzliche Aufwand für die Implementierung des Adapters an.

Der Search Broker überführt die Suchanfrage anhand der Zielsystembeschreibung in von den Zielsystemen verstandene parametrisierte Anfragen (Abbildung 6). Er nutzt hierfür sowohl externe Dienste, wie einen Geonamensdienst (Gazetteer Service), als auch das Ontologiesystem als Quellen, um z.B. den Sachbezug von Suchbegriffen aufzulösen. Das Ontologiesystem muss nicht die gesamte Interpretationsarbeit der Suchparameter leisten, sondern nur noch die semantische Interpretation der Konzeptbezogenen Suchparameter (Abecker et al. 2009a).

Die zu einer Suchanfrage durch das Ontologiesystem ermittelten Ergebnisse sind – mit Hilfe eines dafür definierten XML-Formates - für die Weiterverarbeitung im Portal aufbereitet. Die Struktur umfasst Zielbeschreibungen für Karten, HTML-Seiten bzw. HTML-Fragmente, Medienangebote mit Bezug zum Suchergebnisraum, Referenzen auf konkrete Systeme und Dienstleistungen sowie Anweisungen für die Volltextsuche. Das Portal kann nun Anfragen an die konkreten Zielsysteme erzeugen. Die von den Zielsystemen erhaltenen Resultate umfassen eine Vielzahl von Formaten – je nach System, Inhaltsart und technischen Möglichkeiten des Systems. Sie werden anschließend durch verschiedene Komponenten eines „Mashup“-Steuerungsmoduls inhaltlich aufbereitet und zu einer strukturierten Gesamtergebnisseite zusammengebaut (Abecker et al. 2009a).

3.4.6 Anfragen

Mit dem Ziel, dem Nutzer eine einfache und seinen Gewohnheiten (Nutzung von Internet-Suchmaschinen) entsprechende Nutzerschnittstelle anzubieten, bedeutet knappe, ggf. unspezifische Suchanfragen zu bekommen. Die Suchanfragen müssen ausgewertet, ggf. expliziert und angereichert werden.

Die vom Nutzer eingegebenen Suchbegriffe werden unverändert dem SearchBroker (Abbildung 6) übergeben. Zum Erkennen der Semantik einer Suchanfrage stehen dem SearchBroker eine Reihe spezialisierter Plugins zur Verfügung. Die Plugins analysieren jeweils die Bestandteile (z.B. einzelne Begriffe) der Suchanfrage und versuchen, ihnen eine explizite Semantik zuzuordnen. So können z.B. eines oder mehrere Gazetteer-Plugins den Ortsbezug innerhalb einer Suchanfrage auflösen. Auch viele Internet-Suchmaschinen verwenden solche Gazetteer-Dienste, die von verschiedenen Anbietern online zur Verfügung gestellt werden. Daneben bieten einige Umweltbehörden spezialisierte Gazetteer-Dienste an, die neben Ortsnamen z.B. auch Flurnamen, Gewässernamen oder Namen von Naturräumen auflösen können.

Zum Beispiel kann das Gazetteer-Plugin einen Ortsnamen wie „Karlsruhe“ erkennen, und ihm weitere Eigenschaften zuordnen, wie:

- `geo:commune:name = Karlsruhe`
- `geo:commune:id = 08212000`

Letztere kann nun z.B. als Parameter für die Adressierung eines Zielsystems dienen, das eben solche (standardisierten) Gemeindekennziffern verarbeiten kann.

Ein anderes Gazetteer-Plugin kann zum selben Ortsnamen „Karlsruhe“ weitere Informationen liefern:

- `geo:lon = 8.4037563`
- `geo:lat = 49.0080848`
- `geo:bbox = 8.2756969,48.9494975 8.5318157,49.0666033`

Auf ähnliche Weise können durch ein weiteres Plugin auch zeitliche Begriffe expliziert werden, z.B. einer Suche mit dem Bestandteil „Sommer 2010“ die expliziten Start- und Enddaten zugeordnet werden:

- `datetime:calendar:day:first = 2010/06/21`
- `datetime:calendar:day:last = 2010/09/22`

Die Auflösung eines Themenbezugs stellt die wohl größte Herausforderung bei der Vorverarbeitung dar. Ziel der Auflösung ist die Abbildung des Themenbezugs auf eines oder mehrere Elemente (Konzepte, Deskriptoren) eines wohldefinierten Wortgutes, wie es in den angeschlossenen Zielsystemen verwendet wird. Die thematische Auflösung ist über eine Suche im oben beschriebenen Ontologiesystem realisiert. Die Suche kann insbesondere durch einen Parameter gesteuert werden, der die Größe der Umgebung

(= den maximalen Abstand von Konzepten und Individuen) zu den gefundenen Suchbegriffen vorgibt (Bügel et al. 2011c).

Ergebnis der thematischen Zuordnung ist eine Liste von Konzepten/Deskriptoren mit eindeutigen Bezeichnern inklusive deren Abstand zu den verwendeten Suchbegriffen.

3.4.7 Mashup-Steuerung und Ergebnisdarstellung

Nach Abschluss der Vorverarbeitung kann der SearchBroker auf Basis der Zielsystembeschreibungen entscheiden, für welche Zielsysteme die notwendigen Informationen vorliegen und wie sie angefragt werden können. Die Abfrage von Daten kann nun der SearchBroker selbst vornehmen oder die entsprechenden vollständigen Adressen an das Suchportal zurückliefern. Derzeit geht die Implementierung den zweiten Weg.

Im Umweltportal ist eine Mashup-Komponente für die Darstellung der Suchergebnisse verantwortlich. Sie kann abhängig von den gelieferten Ergebnissen entscheiden, in welcher Form sie dargestellt werden sollen. Für die Darstellungen liegen verschiedene HTML-Schablonen (Vorlagen) vor, die bei Bedarf mit Daten konkreter Datensätzen verknüpft und in die Ergebnisseite eingebaut werden.

Im Wesentlichen wird dabei zwischen folgenden Zielformaten unterschieden:

- Geodaten, z.B. darstellbar in einem Web Map-Client, z.B. Google Maps API
- Listen von Objekten mit Verweisen auf deren Darstellung in einem Fachsystem, z.B. in Form von Linklisten
- Tabellarische Daten und Diagramme, die ggf. nach HTML konvertiert werden
- Multimedia-Inhalte, z.B. in Form einer Galerie-Ansicht
- Text-Nachrichten (Formate Atom oder RSS), z.B. in Form von Übersichtslisten
- HTML-Seiten, HTML-Fragmente und Mikroformate, die an bestimmten Stellen im Layout eingeblendet werden
- Ergebnisse der Volltextsuche, z.B. in Form von Trefferlisten.

Die Zuordnung der Zielformate erfolgt über die in den Zielsystembeschreibungen angegebenen MIME-Typen (Freed und Borenstein 1996). Der Mashup-Generator kann bei mehreren zur Verfügung stehenden Zielformaten auf Basis einer Konfiguration entscheiden, in welcher Form die Treffer angezeigt werden sollen. Dabei können dieselben Daten durchaus an mehreren Stellen, z.B. innerhalb einer Trefferliste und einer Kartendarstellung verwendet werden.

Darüber hinaus bieten die während der Vorverarbeitung gefunden Informationen die Möglichkeit, dem Nutzer im Umweltportal weitere Navigationsschritte anzubieten. Dazu gehören zum Beispiel die Einschränkung bzw. Erweiterung des Suchraums auf Basis von Unter- respektive Oberbegriffen, aber auch die Auflösung von Mehrdeutigkeiten, die sich z.B. aus der Vorverarbeitung von Ortsnamen ergeben.

3.4.8 Bewertung

Die zweite Architekturvariante stellt eine erhebliche Erweiterung gegenüber der ersten Architekturvariante, bei der es sich eher um das „Aufbohren“ einer vorhandenen Suchlösung handelte, und damit eine eigenständige Suchmaschinen-Architektur dar. Sie arbeitet erstmals mit expliziten Vokabularen in Form von Ontologien, gegen welche Suchanfragen und Zielsysteme annotiert und damit expliziert werden. Der Zugriff auf die Zielsysteme wird auf Basis von Zielsystembeschreibungen (OpenSearch Descriptions) dem Suchsystem explizit bekannt gemacht. Die Informationen zum Inhalt der Zielsysteme steckt in Form von Konzeptnamen ebenfalls in den Zielsystembeschreibungen. Der SearchBroker übernimmt die Ablaufsteuerung bei der Suche, orchestriert die Erweiterung bzw. Explizierung der Suchanfrage über Gazetteer-Dienste sowie über das Ontologiesystem, fragt die einzelnen Zielsysteme ab und führt die Ergebnisse zu einem Gesamtsuchergebnis zusammen. Die Darstellung des Suchergebnisses wird Vorlagen-basiert zentral im Suchportal generiert.
Die Harmonisierung der Vokabulare, die alle als Ontologien dargestellt werden, geschieht über eine Artikulationsontologie, die Relationen zwischen passenden Konzepten der anderen Ontologien vorhält.

Erkenntnisse aus der Implementierung und dem probeweisen Betrieb der zweiten Architekturvariante sind:

- Die semantische Einordnung (Annotation) und Erweiterung der Suchanfrage (bzw. der enthaltenen Suchbegriffe) mit Hilfe von Gazetteer-Diensten ist sinnvoll und notwendig.
- Spezialisierte Teilkomponenten (z.B. Gazetteer-Services) für semantische Erweiterung der Suchanfrage sind sinnvoll. Eine Kaskade bzw. Kommunikation dazwischen ebenso, z.B. um aus gewonnen Zusatzinformationen noch weitere generieren zu können (zum Beispiel *Ortsname ⇒ erkannter Ortsbegriff ⇒ Gemeindekennziffer*). Die möglicherweise durch unabhängige Vorverarbeitungskomponenten entstehenden Inkonsistenzen bzw. Unschärfen (z.B. mehrere gefundene Orte) sind häufig hinnehmbar, da am Ende ein menschlicher Nutzer nochmals die Relevanz der ihm angebotenen Informationen überprüft.
- Insbesondere auch, um mögliche Parameter für spezifische Anfragen an spezifische Zielsysteme gewinnen zu können
- Die verwendeten Vokabulare müssen die Zieldomäne (z.B. des Portals) möglichst vollständig abdecken.
- Werden mehrere Teil-Vokabulare verwendet, so müssen deren inhaltliche Schnittstellen miteinander verknüpft werden, d.h. gleichbedeutende Deskriptoren bzw. Konzepte müssen aufeinander abgebildet werden. Das kann z.B. mit Hilfe einer Artikulationsontologie geschehen (Bügel et al. 2011c). Die Erstellung einer Artikulationsontologie lässt sich teilautomatisieren, bedarf jedoch auch redaktioneller Arbeit.
- Derzeit existierende Ontologiesysteme sind in der Regel nicht für die Verarbeitung großer Mengen von Objektinstanzen geeignet. Bereits wenige 100.000 In-

stanzen führen in der Praxis zu nicht akzeptablen, langen Antwortzeiten. Hier ist z.B. eine Vorverarbeitung (Indexierung) notwendig, oder es wird auf die Speicherung von Objektinstanzen innerhalb des Ontologiesystems verzichtet.

- Das Anbinden von bestimmten Zielsystemen ist schwierig, insbesondere wenn keine geeigneten Schnittstellen (Inhalt, Performanz, techn. Format etc.) für deren Abfrage existieren. In solchen Fällen ist es nicht mehr mit der Entwicklung einfacher Adapter getan, sondern es müssen z.B. Daten redundant vorgehalten und z.B. separat indexiert werden. In einer sehr heterogenen Systemlandschaft bedeutet das im schlimmsten Fall, dass je Zielsystem eine komplexe „Adaptersoftware“ zu implementieren ist.
- Die rein serverseitige Aufbereitung von Ansichten (Mashups) führt zu unflexiblen und statisch wirkenden Anwendungen, insbesondere weil Interaktionen (mangels Möglichkeit zu deren Verarbeitung im Client) in der Regel zum kompletten Neulanden einer gesamten Trefferansicht führen. Ein Suchergebnis wird erst dann an den Client ausgeliefert, wenn alle Verarbeitungsschritte abgeschlossen sind, d.h. das gesamte Suchergebnis vorliegt, was zu spürbaren Wartezeiten für den Nutzer führt.
- Die rein serverseitige Verarbeitung der Suche stellt darüber hinaus einen Flaschenhals der gesamten Suchanwendung dar. Es werden erst Ergebnisse ausgeliefert, wenn alle Anfragen abgearbeitet sind, schlimmstenfalls nach einem Timeout.

  Speziell die einzelnen Zielsysteme könnten stattdessen auch asynchron angefragt und die Ergebnisse dynamisch nachgeladen und dargestellt werden. Entsprechende dynamische Mechanismen im Client sind eine notwendige Voraussetzung hierfür.
- Wenn aus den aus der Ontologie ermittelten Strukturen eine Möglichkeit zur Weiternavigation (z.B. Verfeinerung der Suche) gewonnen werden soll, so führen die Navigationsschritte wegen der vollständig serverseitigen Verarbeitung der Suche in der Regel zur einer kompletten Neuanfrage an das System. Sinnvoll wäre dagegen eine Kommunikation zwischen Navigationskomponente und dem Ontologiesystem selbst, z.B. um einzelne Interaktionsschritte (z.B. das Auf- und Zuklappen eines Teilbaumes in einer Navigationshierarchie) unabhängig von einer Suchanfrage behandeln zu können.
- Da einzelne Komponenten zu Anzeige von Suchergebnissen in der Regel ohnehin auf die Verarbeitung bestimmter Datentypen und -formate spezialisiert sind (z.B. eine Diagrammkomponente nur Zeitreihendaten) ist es lediglich notwendig, jeder Komponenten ausschließlich die für sie bestimmten Daten zur Verfügung zu stellen - nicht den gesamten Datenbestand.
- Um Anzeigekomponenten effizient einsetzen zu können, sollen sie möglichst generisch sein, d.h. hochkonfigurierbar und ausgestattet mit möglichst flexiblen und standardisierten Schnittstellen.
- Für die verwendeten Szenarien ist eine Unterscheidung von Thema, Orts- und Zeitbezug sinnvoll. Ein Thema könnte zusätzlich in ein Thema und einen As-

pekt zerfallen, z.B. “Windkraftanlage erzeugt Leistung”. Das Konzept “Windkraftanlage” steht also in Beziehung zu einem weiteren Konzept “Leistung”.

Funktional und inhaltlich stellt die zwei Architekturvariante eine deutliche Verbesserung gegenüber der ersten Architekturvariante dar. Inhalte werden entsprechend ihrer Zielsysteme anhand der Zielsystembeschreibungen semantisch klassifiziert und Suchanfragen werden ebenfalls semantischen Konzepten zugeordnet. Bestimmte Klassen von Suchbegriffen können semantisch erweitert werden und so z.B. geografische Informationen identifiziert und expliziert werden. Ergebnisansichten werden auf Basis von Schablonen passend zu den gefundenen Suchergebnissen generiert.

Den wichtigsten Kritikpunkt an der zweiten Architekturvariante stellt die rein serverseitige Verarbeitung der Suchanfragen inklusive Generierung der Ergebnisansichten dar. Es entsteht ein Flaschenhals, der zu erheblichen Wartezeiten für den Nutzer führen kann. In künftigen Architekturvarianten sollten daher nur die unbedingt notwendigen Schritte auf Serverseite verarbeitet und mehr Aufgaben in den Client verlagert werden, z.B. die Abfrage konkreter Suchergebnisse aus den Zielsystemen bzw. aus Systemen, welche die entsprechenden Daten performant liefern können. Einen entsprechenden Ansatz und damit eine dritte Architekturvariante beschreibt der folgende Abschnitt.

3.5 Dritte Architekturvariante: Serviceorientierung, „Webcache“, clientseitige Verarbeitung

Die dritte Architekturvariante ist in Abbildung 9 dargestellt. Dabei handelt es sich um eine serviceorientierte Architektur (Fröschle und Reinheimer 2007; Erl 2016) mit einer clientseitigen Verarbeitung der Suchanfrage und Generierung der Ergebnisansicht. Die Daten aus den Zielsystemen werden dabei in einer „Webcache“ genannten Infrastruktur über standardisierte Dienste zugänglich gemacht.
Der Begriff „Dienst“ beschreibt ein unabhängiges Softwaresystem, das eine eigenständig nutzbare Funktionalität über klar definierte Schnittstellen zur Verfügung stellt. Klassischerweise wird dabei eine Trennung von Anwendung (Funktionen des Dienstes) und Frontend (Nutzerschnittstelle) erreicht.

3.5.1 Webcache

Wenn Informationen über zentrale Einstiegspunkte wie Webportale und mobile Anwendungen zur Verfügung gestellt werden sollen, müssen die Inhalte, die auf solchen Anwendungen basieren, meist auf Basisdaten zurückgreifen, die zu anderen Zwecken und in anderen Kontexten entstanden sind bzw. erhoben wurden. Der primäre Zweck der (ursprünglichen) Daten widerspricht häufig dem Zweck der Nachnutzung, z.B. der Präsentation in einem Portal. Zum Beispiel enthalten Informationen personenbezogene Daten, sind Lizenzen unterworfen, bestehen aus großen Informationsmengen, sind nur mit den entsprechenden Benutzerrechten zugänglich, werden in speziellen Datenformaten gespeichert, sind nicht über das Internet zugänglich, sind für Laien unverständ-

lich, sind nicht rund um die Uhr verfügbar etc. Das bedeutet, dass die Originaldaten transformiert (z.B. gefiltert, aggregiert, anonymisiert etc.) werden müssen, ehe sie über ein Portal bereitgestellt werden können. Für die Zwecke der semantischen Suche kann die Transformation auch zur Anreicherung der Daten um semantische Informationen verwendet werden.

Die Grundidee des Webcache in der vorlegenden Arbeit ist die Bereitstellung von "internetfähigen" Kopien der Originaldaten (Abbildung 9). Die Informationen werden automatisch aus den ursprünglichen Systemen (Zielsystemen) extrahiert, z.B. Datenbanken und Fachanwendungen, die dann verarbeitet (Transformation) und in redundanten Systemen (dem Webcache) über standardisierte Schnittstellen (APIs) zur Verfügung gestellt werden. Die Vermeidung des direkten Durchgriffs auf die originalen Datenquellen (Zielsysteme) ermöglicht eine bessere Verfügbarkeit und nutzungsbasierte Skalierung von Diensten (Datendienste und Datenmanagement) und bietet Sicherheitsvorteile durch strikte Trennung von internem und externem/öffentlichem Zugang.

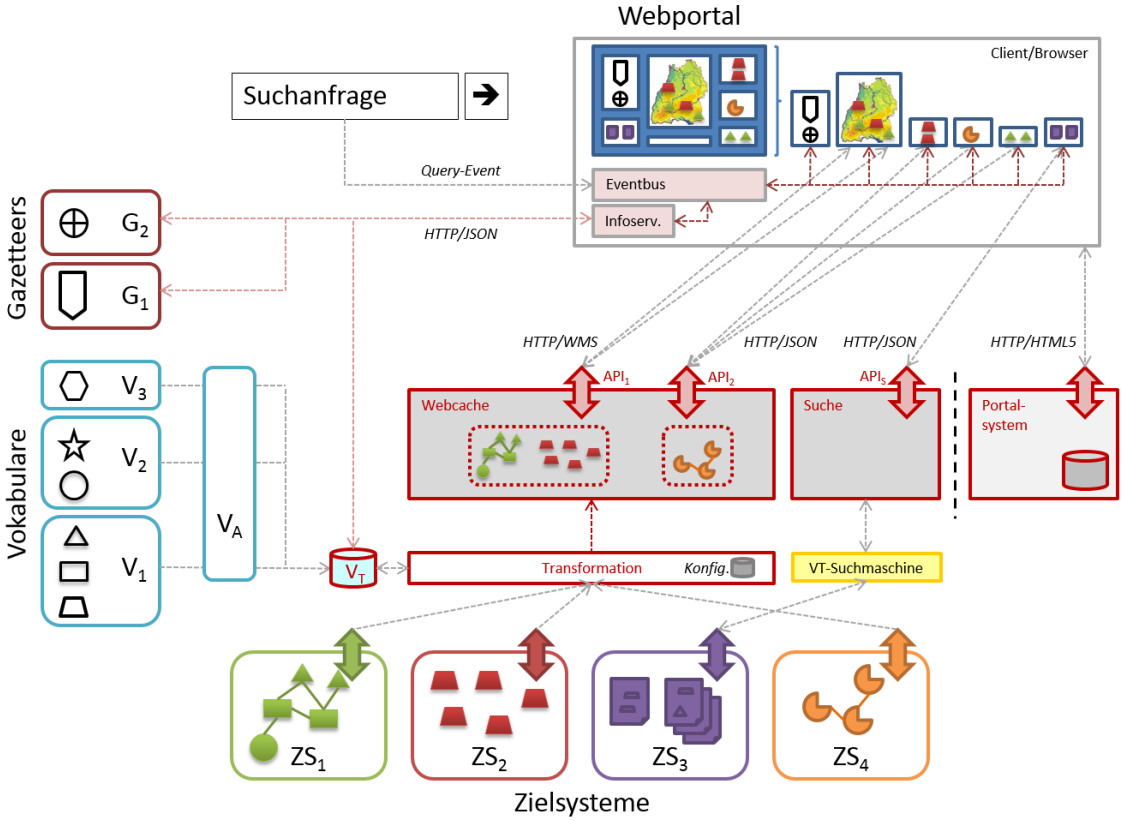

Abbildung 9: Umsetzung als serviceorientierte Architektur mit Aufbau eines „Webcache“ als Sammlung generischer Dienste (neue Entwicklungen und eigene Anteile in rot)

Der Datenfluss ist zunächst unidirektional vom Zielsystem hin zum Webcache konzipiert. So stellt der Webcache eine schreibgeschützte Kopie der Daten dar. Konsistenz- oder Kohärenzbedingungen werden für jeden Datentyp und jedem Zielsystem festgelegt, die die Art und Häufigkeit der Synchronisation zwischen Zielsystem und Webcache beeinflussen. Begrenzt auf unidirektionale Datenflüsse stellt der Webcache

eine vereinfachte Anwendung dar. Im Allgemeinen hat er jedoch die volle Funktionalität des Datenmanagements, d.h. auch Funktionen zum Hinzufügen, Aktualisieren und Löschen von Daten stehen zur Verfügung, einschließlich Mechanismen für Authentifizierung und Autorisierung (Czernik 2016).

3.5.2 Generische Services

Um den Aufwand zur Einrichtung und zum Betrieb des Webcache auf einem akzeptablen Niveau zu halten, besteht ein wesentliches Ziel darin, die gesamte Information durch eine begrenzte Anzahl von generischen Diensten bereitzustellen, in Abbildung 9 angedeutet durch deren Schnittstellen API_i (i = 1...n). Dazu müssen die notwendigen Daten und Funktionalitäten definiert werden. Je nach Anwendung ist neben speziell entwickelten, generischen Diensten auch der Einsatz von Cloud-Services (s. Anhang A5) möglich (Schlachter et al. 2014a).

Zunächst muss die Transformation der Originaldaten in ihre generische Darstellung definiert werden. Das kann eine weitere Verarbeitung wie Auswahl, Filterung, Aggregation etc. (Datenaufnahme, eng. Data Ingestion) umfassen. Der gewählte Ansatz hat den Vorteil, dass nur zulässige "internetfähige" Daten den Webcache erreichen können, z.B. ohne personenbezogene Daten, unter Verwendung aggregierter Werte, Filterung etc.

Für die Anforderungen im Umwelt- und Energiebereich wurden insgesamt acht generische Dienste identifiziert:

- Stammdatendienst
- Schemadienst
- Zeitreihen-Dienst
- Mediendienst
- (Volltext) Suchdienst
- Geodatendienst
- Metadatendienst
- Verbindungsdienst.

Die acht Kerndienste werden durch zwei weitere Dienste ergänzt, die konsumierende Anwendungen unterstützen:

- Anwendungskonfigurationsdienst
- Discovery-Dienst.

Alle Dienste sind weitgehend voneinander unabhängig, eine zentrale Anforderung an den für Microservices verwendeten Komponentenbegriff (Fowler und Lewis 2015). Daher ist eine Implementierung unter Verwendung von Microservices offensichtlich. In Laufzeitcontainern wie Docker (Docker 2016; Mouat 2016) verpackt, können sie ohne zusätzlichen Aufwand an einer Vielzahl von möglichen Infrastrukturen wie dedizierten Servern, Clustern oder in der Cloud betrieben werden. Unter Einsatz von Laufzeit-Infrastrukturen wie Kubernetes (Kubernetes 2016) sind operative Aspekte wie Rolling-

Updates, Monitoring, Skalierbarkeit und Load-Balancing nur eine Frage der Konfiguration - eine geeignete Computerinfrastruktur und Software-Design vorausgesetzt.

Alle Dienste nutzen geeignete Backend-Systeme, die insbesondere die Persistenz der Daten sicherstellen. Auch hier wird die Architektur von konkreten Systemen abstrahiert, so dass die Backend-Systeme problemlos ausgetauscht oder gleichzeitig unterschiedliche Backend-Systeme genutzt werden können. Die Auswahl geeigneter Backend-Systeme, z.B. verschiedene NoSQL-Technologien (Edlich 2011; Sullivan 2015), sichert auch dynamische Eigenschaften wie Lastverteilung, Skalierbarkeit etc.

Alle Services bieten ihre Funktionalität durch versionierte RESTful-Interfaces (Fredrich 2013) über Content-Negotiation (W3C 1999), d.h. der Client kann bei der Anfrage das gewünschte Rückgabeformat angeben, bzw. eine Prioritätenliste, sofern mehrere Rückgabeformate verfügbar sind. Das erleichtert die Entwicklung, Wartung und den Austausch einzelner Dienste.

Die postulierte Unabhängigkeit der Dienste darf jedoch nicht zu einem Verlust an möglicher Funktionalität führen, zum Beispiel durch fehlende Interoperabilität. Daher stellt die Architektur grundsätzlich eine Messaging-Infrastruktur (Channels) bereit (Fowler und Lewis 2015; Olliffe 2015). Jedoch kann die Messaging-Schicht „unter“ den Microservices nicht wie die Gartner-Microservice-Architektur (Olliffe 2015) an die Datenaufnahme und die Nutzung durch Anwendungen unter Verwendung eines Ereignisbusses delegiert werden, eine Vereinfachung, die für Anwendungsfälle mit unidirektionalem Datenfluss ausreicht.

3.5.3 Generische Frontend-Komponenten

Für die Präsentation von Daten und Inhalten in Nutzer-Frontends, z.B. Webportalen, Websites oder mobilen Anwendungen sind zur Darstellung verschiedener Datenformate entsprechende Frontend-Komponenten notwendig, z.B. zur Anzeige von

- Objektdaten (z.B. als Einzelansicht, als Liste oder als Tabelle),
- Diagrammen (z.B. Zeitreihen, Tortengrafiken),
- Geoinformationen auf einer Karte,
- Auswahllisten (z.B. zur Nachfilterung anhand von Facetten),
- Formularen.

Insbesondere innerhalb von Portalen, die per se Daten aus verschiedenen Quellen darstellen sollen, ist es sinnvoll, die Komponenten generisch, d.h. konfigurierbar und damit nutzbar für verschiedene Anwendungsfälle, auszulegen. Das bedeutet, dass z.B. die Datenquelle (Schnittstelle, Format), das Ausgabeformat (z.B. in Form von Templates), spezifische Styling-Formate (z.B. Cascading Stylesheets, CSS) und weitere Einstellungen per Konfiguration für eine spezifische Instanz der Komponente festgelegt werden können.

Generische Frontend-Komponenten sollten darüber hinaus flexibel in verschiedene Basissysteme integrierbar sein, zum Beispiel Content-Management-Systeme oder Enterprise Portal-Systeme.

Im Bereich des Ministeriums für Umwelt, Klima und Energiewirtschaft des Landes Baden-Württemberg werden seit 2014 viele Webportale auf Basis der Portal-Software Liferay Portal (Community Edition) (Liferay 2014) realisiert. Auf den durch die Portalsoftware generierten Seiten können individuelle Bausteine (Frontend-Komponenten) verwendet werden. Der für die Anordnung von Frontend-Komponenten innerhalb von Liferay verwendete Mechanismus entspricht dem Portlet-Standard JSR-286 („Portlet 2.0“) (Liferay 2016). Frontend-Komponenten müssen für die sinnvolle Nutzbarkeit in Liferay also mindestens dem JSR-286-Portlet-Standard entsprechen. Um jedoch eine weitergehende und von Liferay bzw. dem Portlet-Standard unabhängige Wiederverwendbarkeit gewährleisten zu können, setzen alle Frontend-Komponenten auf grundlegenderen Technologien auf. Sie sind als Web Widgets (Lal und Chava 2010) realisiert, d.h. unabhängige JavaScript-Programme, welche ihre Ausgaben in der Regel in Form von HTML-Code in die Seite injizieren, in der sie enthalten sind. Alle Web-Widgets sind dabei, wie oben beschrieben, auf eine generische Nutzbarkeit bzw. Mehrfachverwendung ausgelegt und hochgradig parametrisierbar. Zusätzlich haben alle Widgets die Möglichkeit über einen Eventbus Nachrichten auszutauschen bzw. sich über bestimmte Ereignisse informieren zu lassen, z.B. wenn sich der geographische Kontext der Anzeige ändert. Generische Widgets lassen sich daher in der Regel ohne Modifikation in verschiedenen Portalen wie Liferay, CMS-Systemen wie Typo3 (Typo3 2014) oder WebGenesis (Fraunhofer IOSB 2016) und sogar in statischen HTML-Seiten nutzen.

Die Verwendung von Web Widgets resultiert aus der Notwendigkeit zur Abwärtskompatibilität zu älteren Webbrowsern. Sobald eine breite Browser-Unterstützung für den neuen HTML5-Standard „Web Components“ gegeben ist, sollen die Widgets darauf umgestellt werden. Die wesentlichen Vorteile von Web Components gegenüber Web Widgets betreffen ihre Unabhängigkeit vom weiteren Inhalt der HTML-Seite, z.B. was die Mehrfachverwendung einer Komponente innerhalb einer Seite, die Auswirkungen von CSS-Stylings, die Ablage von Daten innerhalb der Seite sowie weitere mögliche Wechselwirkungen (Plugins, JavaScript-Code etc.) betrifft – zusammengefasst: Web Components sind besser gekapselt als Web Widgets (W3C 2016; Lal und Chava 2010). Dazu tragen vor allem die unter dem Titel Web Components definierten Standards für benutzerdefinierte HTML-Tags, Imports von HTML-Fragmenten, HTML-Templates sowie die Abgrenzung von Teilen des HTML-Dokumentes per Shadow DOM bei (webcomponents.org 2016).

Zur komfortablen Nutzung innerhalb von Liferay werden sämtliche Widgets in sogenannten Wrapper-Portlets verpackt, d.h. Portlets, die lediglich einen Rahmen für die enthaltenen Web Widgets darstellen, um die Web Widgets als Portlets nutzen und konfigurieren zu können. Die Wrapper-Portlets können durch Autoren entsprechend einem gewählten Layout bequem per Drag&Drop neben weiteren Portlets in Seiten platziert

werden. Darüber hinaus bietet der Portlet-Standard die Möglichkeit, die Konfiguration des Portlets per Web-Oberfläche vorzunehmen. Viele Konfigurationsaufgaben, die früher von Administratoren z.B. im Quellcode oder in Konfigurationsdateien erledigt werden mussten, können so ohne tiefes technisches Wissen durch Redakteure bzw. Web-Autoren durchgeführt werden.

Es stehen Widgets für die Darstellung von raumbezogenen Inhaltsobjekten auf einer Karte auf Basis der Google Maps API (Kartenwidget), zur Auswahl von Kartenlayern, zur Selektion von Orten, zur Abfrage und zur Darstellung von Messdaten, mehrere Widgets zur Darstellung von Trefferlisten aus der Volltextsuche, zur Darstellung von Sachdaten (Listen von Objekten oder Facettierungsinformationen), zur Darstellung von Objektdaten (Einzelobjekte), zum Anzeigen von Fotos/Bildern, zur Darstellung von Veranstaltungslisten, Kalendern sowie von News-Einträgen (z.B. RSS-Feeds) zur Verfügung. Die meisten Portlets/Widgets nutzen dabei die im vorigen Abschnitt beschriebenen Dienste. Darüber hinaus können auch allgemein verfügbare Liferay-Portlets verwendet werden.

3.5.4 Zusammenspiel von Frontend-Komponenten

Aus Sicht eines Nutzers ergibt sich eine Gesamtanwendung meist aus dem Zusammenspiel mehrerer Einzelteile. Im Falle der Landesumweltportale besteht die Gesamtanwendung aus der Orchestrierung mehrerer Frontend-Komponenten respektive der ihnen zugrundeliegenden Dienste. Die Orchestrierung ergibt dann beispielsweise eine Suchergebnisseite, die Informationen aus gleichen oder verschiedenen Quellen auf unterschiedliche Weisen visualisiert.

Im Falle der Suchergebnisseite werden die anzuzeigenden Ergebnisse meist durch die manuelle Eingabe von einem oder mehreren Suchbegriffen getriggert.

In einem Vorverarbeitungsschritt wird versucht, den Suchbegriffen eine Semantik zuzuordnen, z.B. thematische Begriffe wie „Bodenerhaltung“ auf bekannte Deskriptoren wie „Bodenschutz“ abzubilden oder geographischen Begriffen wie „Karlsruhe“ den entsprechenden Ort zuzuordnen. Bei der Vorverarbeitung können die Begriffe semantisch angereichert werden, z.B. dem erkannten Ortsnamen „Karlsruhe“ auch dessen geografischer Mittelpunkt (Center), eine Bounding-Box sowie ein amtlicher Gemeindeschlüssel (Gemeindekennziffer); thematische Begriffe können ebenfalls um Attribute ergänzt werden, z.B. Schlüssel wie ein Objektartencode oder ein Fachführungscode. Mehrdeutigkeiten, beispielsweise mehrere erkannte Ortsnamen zu einem Suchbegriff wie „Neuhausen“, können ggf. erst nach einer Nutzerinteraktion aufgelöst werden.

Der Kontext einer Abfrage kann auch durch die Anwendung ergänzt werden, z.B. einen durch den Nutzer vorgegebenen bevorzugten Standort, der im System oder einem Cookie gespeichert ist und allen Suchanfragen automatisiert hinzugefügt wird.

Das wird in Abbildung 10 anhand des Umweltinformationsnetzes Sachsen-Anhalt veranschaulicht. Hier wurde nach den Suchbegriffen „hochwasser tangermünde“ gesucht. Neben einer klassischen Volltextsuche-Trefferliste (rechts unten) sind hier unter ande-

rem eine Kartenansicht (rechts oben) mit der zugehörigen Kartenlayerauswahl (links oben) sowie passende Treffer konkreter Objekte (darunter, z.B. „Risiko (407)“) sowie aus dem Metadatenkatalog des Landes Sachsen-Anhalt (links unten) zu sehen.
Die inhaltliche Auswahl der angezeigten Ergebnisse geschieht auf Basis der Vorverarbeitung der Suchanfrage, z.B. erkennt der Geo-Gazetteer-Dienst den Ortsnamen „Tangermünde“ und die liefert die entsprechenden Geokoordinaten bzw. Bounding-Box, die z.B. genutzt werden, um den Kartenausschnitt auf die Gemeinde zu zentrieren. Die Auswahl der Kartenlayer geschieht durch den Abgleich des erkannten Konzeptes „Hochwasser“ mit den in den Konfigurationen der Kartenlayer abgelegten Konzepten. Kommt es dabei zu einem direkten Treffer, wird der entsprechende Kartenlayer direkt angezeigt, z.B. „Hochwassergefährdung HQ100“ oder „Wassererlebnis“. Verwandte Themen werden zwar in der Layerauswahl angezeigt, müssen jedoch zur Anzeige durch den Nutzer interaktiv ausgewählt werden, z.B. „Badegewässer“.

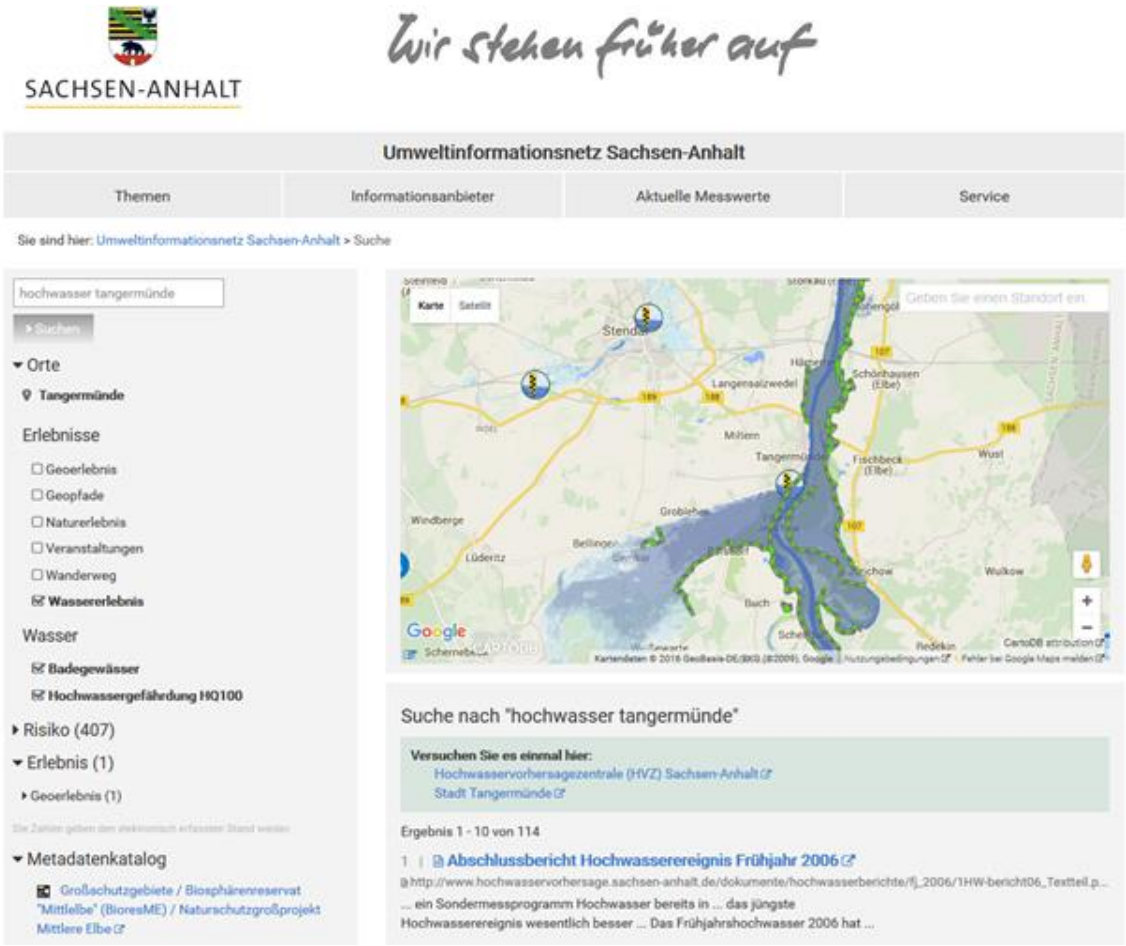

Abbildung 10: Suchergebnisseite mit Karte, Layer-Auswahl, Volltext- und Metadaten-Trefferlisten im Umweltinformationsnetz Sachsen-Anhalt (Screenshot Umweltinformationsnetz Sachsen-Anhalt)

Die Verarbeitung und Verbreitung der Informationen geschieht in den Landesumweltportalen bzw. den dort verwendeten Frontend-Komponenten mit Hilfe einer Ereignisbasierten Kommunikationsschicht, die Eventbus genannt wird. Der Eventbus bietet eine generische Schnittstelle zum Versenden und Empfangen von Nachrichten an.

Jede Komponente kann sich für bestimmte Typen und ggf. bestimmte Absender von Nachrichten registrieren. In Abbildung 10 sendet z.B. die Suchschlitz-Komponente (oben links) die vom Nutzer eingegebenen Suchbegriffe auf den Eventbus. Die Komponente zur Anzeige von Volltextsuchergebnissen (unten links) und die Komponente zur Auswahl von Kartenlayern werten das Event aus und zeigen selbständig passende Volltextsuchergebnisse an bzw. bieten dem Nutzer passende Kartenlayer zur Auswahl an. Die Auswahl eines Kartenlayers löst dann erneut ein Event aus, das der Kartenkomponente (oben rechts) signalisiert, den entsprechenden Layer anzuzeigen.

Auch wenn der Nachrichtenmechanismus grundsätzlich generisch ist, d.h. beliebige Typen von Nachrichten und beliebige Nutzinhalte zulässt, müssen deren Typen sowie die Struktur der Inhalte von Nachrichten durch die Anwendung definiert und von den Komponenten interpretiert werden. Das bedeutet im Falle der Umweltportale, dass sich der Nachrichtenaustausch auf die dort vorhandenen Use-Cases bezieht: Austausch von Suchbegriffen, thematischen Begriffen, Ortsangaben, selektierten Objekten bzw. Objektklassen sowie die An- bzw. Abwahl von Kartenlayern.

Insbesondere die Kommunikation per Ortsangabe bzw. thematischen Begriffen stellt zwar eine relativ lose Kopplung dar, sie kommt in vielen Fällen jedoch der Datenlage insofern entgegen, dass ein Zusammenhang zwischen zwei örtlich benachbarten Objekten (Windkraftanlage in der Nähe eines Naturschutzgebietes) häufig in den Daten nicht explizit dargestellt ist.

Für die Landesumweltportale hat die Event-basierte Kommunikation mit einer losen Kopplung von (Umwelt-)Objekten einen entscheidenden Vorteil. Sie reduziert den Aufwand bei der Einbindung von Umweltdaten in die Landesumweltportale auf ein leistbares Niveau, d.h. Beziehungen zwischen Objekten müssen nicht unbedingt explizit modelliert werden. So kann beispielsweise der in den Daten vorhandene Ortsbezug genutzt werden, um aus der räumlichen Nähe von Objekten eine Beziehung abzuleiten. Die Landesumweltportale bieten auf der einen Seite Zugang zu einer äußerst heterogenen Landschaft von Umweltdaten: strukturierten, semistrukturieren und unstrukturierten Daten, Daten mit und ohne expliziten Ortsbezug in einer Vielzahl von Repräsentationen, Daten aus unterschiedlichen technischen Systemen mit einer Vielzahl von Schnittstellen und technischen Formaten, verschiedenen IDs oder Schlüssellisten etc. – oder mit anderen Worten: Die Daten- bzw. Systemlandschaft bietet in den meisten Fällen keine expliziten Beziehungen zwischen Daten und Objekten bzw. zumindest keine technisch nutzbare Umsetzung dafür.

Auf der anderen Seite erwarten die Nutzer der Umweltportale in den meisten Fällen zwar eine Unterstützung beim Auffinden der passenden Informationen zu ihrem Anliegen, stellen dabei aber selbst eine aktive Filterinstanz dar, welche die angezeigten Informationen sichten, bewerten und sich passende Teile herauspicken kann. Eine – nicht zu große – Obermenge der tatsächlich relevanten Ergebnisse ist für sie in den meisten Fällen akzeptabel. Des Weiteren hat sich gezeigt, dass bei einem großen Anteil der Suchanfragen die Beziehungen zwischen den passenden Ergebnissen auf einem sehr hohen Abstraktionsniveau darstellbar sind, z.B. ihrer örtliche Nähe zueinan-

der. So kann beispielsweise die Frage, ob die Windkraftanlagen einer Gemeinde innerhalb oder in der Nähe von Naturschutzgebieten liegen, mit Hilfe der Umweltportale sehr leicht beantwortet werden (Abbildung 11): Allein durch Eingabe der Suchbegriffe „windrad schutzgebiet langenburg“ erhält der Nutzer bereits die gewünschten Informationen, allerdings tatsächlich mehr als verlangt, da neben Naturschutzgebieten auch Objekte anderer Schutzgebietstypen (Biotope, Nationalparke etc.) dargestellt werden. Durch das einfache Abwählen der nicht benötigten Schutzgebietstypen lässt sich die Frage klären, denn alle Naturschutzgebiete und Windkraftanlagen im Bereich der Gemeinde Langenburg werden angezeigt. Zwar muss der Nutzer die Beziehung zwischen Windkraftanlagen und Schutzgebieten noch selbst herstellen, kann sie dank der Kartenansicht „auf einen Blick“ erkennen – obwohl in der verwendeten Datengrundlage eine Beziehung wie „liegt in/bei“ nicht vorhanden ist; im Gegenteil: Informationen über Windkraftanlagen und Naturschutzgebiete kommen aus völlig unterschiedlichen Systemen und sind nur über die gemeinsame Darstellung innerhalb des Kartenclients im Umweltportal miteinander verbunden.

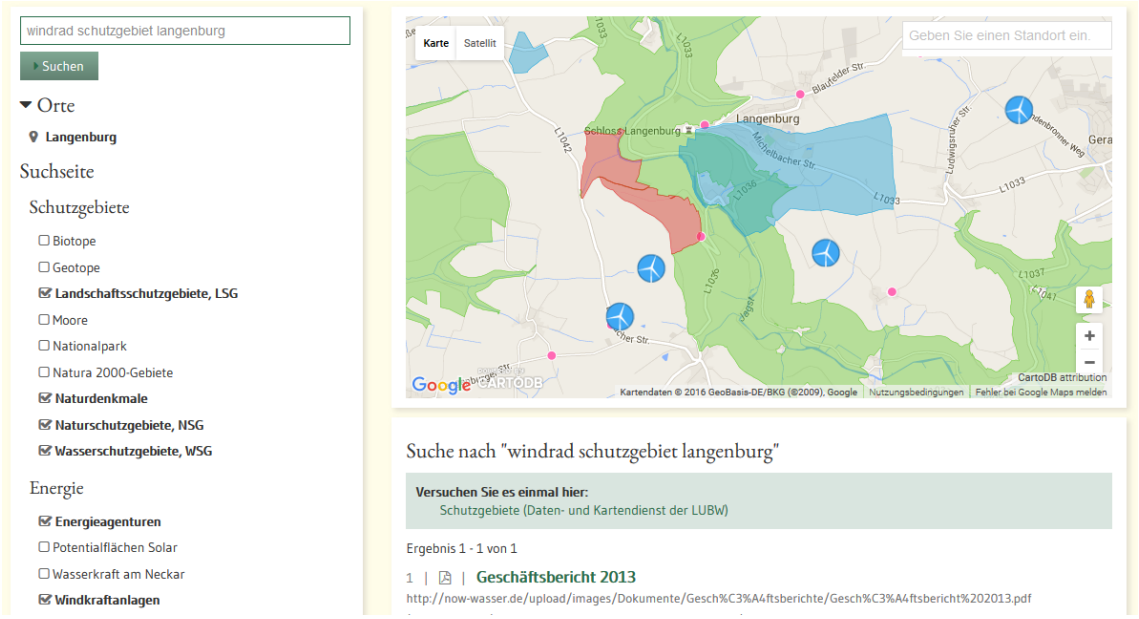

Abbildung 11: Verknüpfung von Windkraftanlagen und (Natur-)Schutzgebieten durch die Suche nach „windrad schutzgebiet langenburg“ im Umweltportal Baden-Württemberg (Screenshot)

Die lose Kopplung von Umweltobjekten funktioniert also nur für menschliche Nutzer und nur in solchen Anwendungsfällen, in denen sich Beziehungen auf relativ hohen Abstraktionsniveaus darstellen lassen, z.B. ihre örtliche Nähe oder die Zuordenbarkeit zu einem bekannten Thema (Windkraft).

3.5.5 Verknüpfung semantischer Objekte und Klassen

Die Umweltportale haben in den meisten Fällen keinen oder nur wenig Einfluss auf die Systeme, aus denen sie ihre Daten beziehen. Das bedeutet, dass Erweiterungen des

Datenmodells, z.B. um Beziehungen zu anderen Objekten, in solchen Systemen in der Regel nicht möglich sind. Um dennoch eine engere Kopplung der Daten aus den Systemen zu erreichen, bietet es sich an, die Beziehungen in zusätzlichen Systemen („Beziehungsdiensten“ oder „Link-Services“) abzulegen. Dort kann eine Beziehung als Tripel (Objekt A, Typ der Beziehung, Objekt B) gespeichert werden, z.B. („Windkraftanlage Nr. 4711“, „liegt-in“, „Gemeinde Nr. 08127047“), was der Mechanik des Semantic Web (RDF-Tripel) entspricht (W3C 2004c). Speichert man zusätzlich in einem Metadatensystem, in welchem konkreten System die Windkraftanlagen gespeichert sind und wie technisch darauf zugriffen werden kann (z.B. Service-Adresse und Schlüsselattribut), dann kann ein Service bereitgestellt werden, der seinerseits explizite Verknüpfungen zu anderen konkreten Objekten (z.B. einzelnen Naturschutzgebieten) bereitstellt, selbst wenn die Beziehungen in der Original-Datenquelle nicht vorhanden sind.

Ein Problem ist allerdings der Aufwand für die Erfassung der Beziehungen zwischen (Einzel-)Objekten. In vielen Fällen lässt sich der Aufwand für die Erzeugung von Beziehungen jedoch reduzieren, da die Verknüpfung häufig nicht auf der Basis von Einzelobjekten erfolgen muss, sondern z.B. eine ganze Klasse von Objekten, z.B. alle Windkraftanlagen, mit einer Eigenschaft verknüpft („Windkraftanlage“, „produziert“, „Energie, elektrisch“) wird. Die Anzahl der Beziehungen von *n* Windkraftanlagen zu einer Eigenschaft oder einem anderen Objekt reduziert sich dabei signifikant von *n* auf 1. Noch drastischer fällt die Reduktion aus, wenn auf beiden Seiten Klassen von Objekten stehen, z.B. *n* Windkraftanlagen und *m* Schutzgebiete, wodurch sich die Anzahl der Beziehungen von *n***m* auf 1 reduziert. Auch wenn Beziehungen wie („Windkraftanlage“, „steht potenziell in Konflikt mit“, „Naturschutzgebiet“) keine Aussagen über das einzelne Objekt treffen, stellen sie dennoch wertvolle Verbindungen zwischen ggf. sonst isolierten Objektklassen dar.

Damit entstehen insgesamt drei Arten von Beziehungen:

- Objekt : Objekt (n:m)
- Klasse : Objekt (1:n)
- Klasse : Klasse (1:1).

Derzeit werden ein Konzept sowie ein erster Prototyp eines solchen „Link-Services“ erarbeitet, welcher die Möglichkeiten für eine engere Kopplung schaffen kann, so dass künftig auch weitergehende Anwendungsfälle abgedeckt werden können und auch maschinelle Nutzer als Konsumenten der Informationen aus den Umweltportalen in Frage kommen. Potenziell sind dann auch weitere Methoden des Semantic Web (z.B. „Reasoning“, d.h. die Generierung neuer Erkenntnisse) möglich. Mit Hilfe von Regeln lassen sich aus der Beziehung zwischen zwei Klassen auch Beziehungen zwischen einzelnen Objekten der Klassen automatisiert generieren. Der Link-Service ist nicht mehr Bestandteil der vorliegenden Arbeit.

3.5.6 Bewertung

Die in den vorigen Abschnitten vorgestellte dritte Architekturvariante bricht mit dem bisherigen Prinzip, Daten und Informationen direkt in den Zielsystemen abzufragen. Stattdessen werden die Daten redundant mit Hilfe einer Sammlung von Datendiensten („Webcache“) bereitgestellt. Aus den Zielsystemen werden sie über einen Transformationsmechanismus in die Backend-Systeme der Datendienste überführt und dabei gleichzeitig mit den notwendigen semantischen Informationen, d.h. den zugehörigen semantischen Konzepten, versehen. Es entsteht eine serviceorientierte Architektur (SOA), die der Client zur Abfrage von Daten im Sinne einer Suche nutzen kann. Auch flankierende Dienste wie der Geo-Gazetteer stellen ihre Funktionalität innerhalb der SOA zur Verfügung.
Die Orchestrierung der Suche im Client geschieht über die Kommunikation der generischen Komponenten, die sich jeweils auf die Abfrage und Anzeige bestimmter Datentypen beschränken.

Einzelne (Micro-)Services sind sehr gut geeignet, um den Zugriff auf Daten und Objekte von ihren jeweiligen Zielsystemen zu entkoppeln. In ihrer Gesamtheit bilden sie eine einheitliche Zugriffsschicht, die über generische Dienste realisiert werden kann und die potenziell große und heterogene Anzahl von Schnittstellen der Zielsysteme erheblich reduziert, was den Zugriff durch ebenfalls generische Frontend-Komponenten sehr vereinfacht.

Aus den bisherigen Untersuchungen ergeben sich die folgenden Aussagen:

- Die Entkopplung der Zugriffsschicht von den Zielsystemen reduziert die Last auf die Quellsysteme auf ein für die Synchronhaltung der Daten notwendiges Minimum. Gleichzeitig kann die Laufzeitinfrastruktur für einzelne Services individuell gewählt und konfiguriert werden, z.B. um eine lastabhängige horizontale Skalierbarkeit einzelner Services gewährleisten zu können.
- Die Entkopplung der Zugriffsschicht (API) von den Quellsystemen sorgt für stabile Schnittstellen, auch wenn sich Schnittstellen oder Datenmodelle von Quellsystemen ändern. Anpassungen müssen teilweise nur für die Anbindung des Quellsystems an den Service vorgenommen werden, massive Änderungen oder Erweiterungen, z.B. am Datenmodell, können über eine Versionierung der Zugriffsschicht (API) abgefangen werden, die eine Rückwärtskompatibilität gewährleisten kann.
- Der Aufbau einer Zugriffsschicht mit eigener, redundanter Datenhaltung erzeugt zunächst Aufwand,
    - Notwendigkeit zur Bereitstellung und zum Betrieb weiterer Systeme (Services),
    - Redundante Datenhaltung und daher die Notwendigkeit eines Konzepts (Regeln, Definition eines Konsistenzbegriffes für potenziell jede einzelne Datenquelle) und eines Systems zum Sicherstellen von Datenkonsistenz,

  - Notwendigkeit zur Konzeption und Implementierung entsprechender Synchronisationsmechanismen,

  bietet jedoch auch sehr gute Chancen:

  - Einheitliche und stabile Zugriffsschicht (APIs, ggf. versioniert) erleichtert die Implementierung von generischen Frontend-Komponenten und weiteren Anwendungen. Damit wird auch das Potenzial für Synergien, d.h. Mehrfachnutzung von Daten für verschiedene Zwecke, erhöht.
  - Bereitstellung der Daten über standardisierte URLs (z.B. RESTful) als notwendige Voraussetzung für Linked Data.
  - Optimierte Systeme für den Zugriff, z.B. Nutzung von Suchmaschinen wie Elasticsearch (Elastic 2016).
  - Vorverarbeitung der Daten bei der Synchronisation, z.B. Erweiterung um semantische Angaben wie Konzepte, Schlagworte etc., Normalisierung von Attributen (Datumsangaben, Geokoordinaten, …), Abbildung auf standardisierte oder generische Attribute (Titel, Kurzbeschreibung, Link etc.), potenziell auch die Erweiterung um Referenzen zu verknüpften Objekten.
- Die Wahl geeigneter Mechanismen (Container) gewährleistet einen flexiblen Betrieb, z.B. auf dezidierten Servern, auf Rechnerclustern oder in der Cloud und ermöglicht einen automatisierten Build- und Deploymentprozess.
- Nicht vollständig instrumentierte Tool-Unterstützung (Ziel: Einheitlichkeit, Konfiguration vor Programmierung) für die Synchronisation und die Sicherstellung der Datenkonsistenz.
- Objekte sind zwar per eindeutiger URL adressierbar, die Beschaffenheit der URLs entspricht jedoch noch nicht in allen Fällen den Best Practises für den Aufbau von RESTful URLs (Fredrich 2013).
- Die durch die Services unterstützten Formate sind zwar standardisiert und maschinenlesbar (z.B. GeoJSON), es wird jedoch z.B. noch kein RDF unterstützt.
- Die Daten enthalten noch keine expliziten Verweise (Links) auf andere Objekte im Sinne von Linked Data.
- Die verwendeten Vokabulare sind nicht formalisiert. De facto werden zwar in den meisten Fällen Deskriptoren des Semantic Network Service (SNS) (Umweltbundesamt 2016) verwendet, hier jedoch die Labels (Zeichenketten) statt der eindeutigen IDs. In den Beschreibungen von Inhalten (z.B. Metadaten zu einem Kartenlayer) werden die Labels des Deskriptors sowie (in der Regel) die Labels eine Reihe von verwandten Begriffen verwendet. Formal sollte jedoch ein Dienst zur thematischen Zuordnung einer Suchanfrage zu bestimmten Deskriptoren bzw. Konzepten verwendet werden, wie ihn z.B. die autoClassify-Funktion des SNS anbietet.
- Die Versionierung der Zugriffsschicht ist noch nicht umgesetzt.

- Die Orchestrierung von Services erfolgt derzeit ausschließlich in den konsumierenden Anwendungen. Es liegen noch keine Erfahrungen in der Inter-Service-Kommunikation vor.

Die dritte Architekturvariante zeigt bereits eindrucksvoll das Potenzial einer semantischen Suche in heterogenen Informationssystemen innerhalb des Umweltinformationssystems Baden-Württemberg. Die Implementierung als serviceorientierte Architektur in Kombination mit der Verwendung moderner HTML5-basierter Webtechnologie zur Implementierung der Client-Komponenten löst das Hauptproblem der zweiten Architekturvariante, den durch die serverseitige Verarbeitung der Suchanfrage entstandenen Flaschenhals.
Die redundante Datenhaltung im „Webcache“ erfordert zwar einen Mehraufwand gegenüber der direkten Abfrage von Daten aus den Zielsystemen, z.B. den Betrieb der Services inklusive der zugehörigen Backend-Systeme, jedoch reduziert sich hierdurch die notwendige Anzahl der durch den Client zu implementierenden Schnittstellen – und damit dessen Komplexität – deutlich. Das nach wie vor notwendige Umgehen mit der Heterogenität der Zielsysteme verschiebt sich hin zum Transformationsprozess (Data Ingestion), der die Daten aus den heterogenen Zielsystemen in den homogeneren Webcache mit seiner überschaubaren Anzahl von Schnittstellen überführt.
In der folgenden vierten Architekturvariante soll die dritte Variante dahingehend erweitert werden, dass die Daten aus dem Webcache auch im Sinne des Semantic Web (Berners-Lee et al. 2001) über standardisierte Schnittstellen als Linked Data (Berners-Lee 2006) bereitgestellt werden können.

3.6 Vierte Architekturvariante: Ausbau zu semantischen Diensten / Linked Data

Die Repräsentation von Daten mit Hilfe von RESTful Webservices (Bayer 2002) bietet gute Chancen, die Daten in Form von Linked Data zur Verfügung stellen zu können. Dabei sind nach Tim Berners-Lee (Berners-Lee 2006) vier Regeln zu beachten:

1. Verwendung von URIs zum Benennen von Objekten
2. Verwendung von HTTP-URIs zum tatsächlichen Auffinden von Objekten
3. Verwendung von Standards wie RDF und SPARQL
4. Bereitstellung von Links in Form von URLs, um Verbindungen zu weiteren Objekten finden zu können.

Wenn alle verfügbaren Daten (Objekte) bereits über (Micro-)Services mit RESTful URIs bereitgestellt werden, können an die ersten beiden Regeln bereits Haken gemacht werden.

Die Bereitstellung der Daten in Standardformaten wie RDF ist dann nur noch eine Frage der formalen Repräsentation, die relativ leicht zu implementieren ist. Wesentliche Randbedingung zur globalen Nutzbarkeit der Daten ist allerdings deren Anbindung an

globale Vokabulare, d.h. deren direkte Nutzung bzw. ein Mapping darauf, um eine Interoperabilität mit weiteren Datenquellen gewährleisten zu können.

Die Verknüpfung von Daten mit weiteren Objekten, d.h. die Bereitstellung von expliziten Links, stellt die größte Herausforderung dar, da in vielen Datenquellen solche Verknüpfungen nicht bestehen, schon gar nicht in Form von URIs. Verknüpfungen zwischen Objekten müssen also zunächst erzeugt werden. Die notwendigen Regeln bzw. zugehörigen Operationen zur Generierung von Verknüpfungen müssen entweder aus den Daten selbst bzw. den zugehörigen Metadaten des Quellsystems gewonnen werden, oder - im schlimmsten Fall - neu erzeugt werden, was in der Regel Aufwände erzeugen wird.

Ziel der vierten Architekturvariante ist die Bereitstellung aller Daten als Linked Data, was die explizite Bereitstellung von Verknüpfungen zwischen Daten beinhaltet. Abbildung 12 zeigt die vierte Architekturvariante mit der entsprechenden Erweiterung.

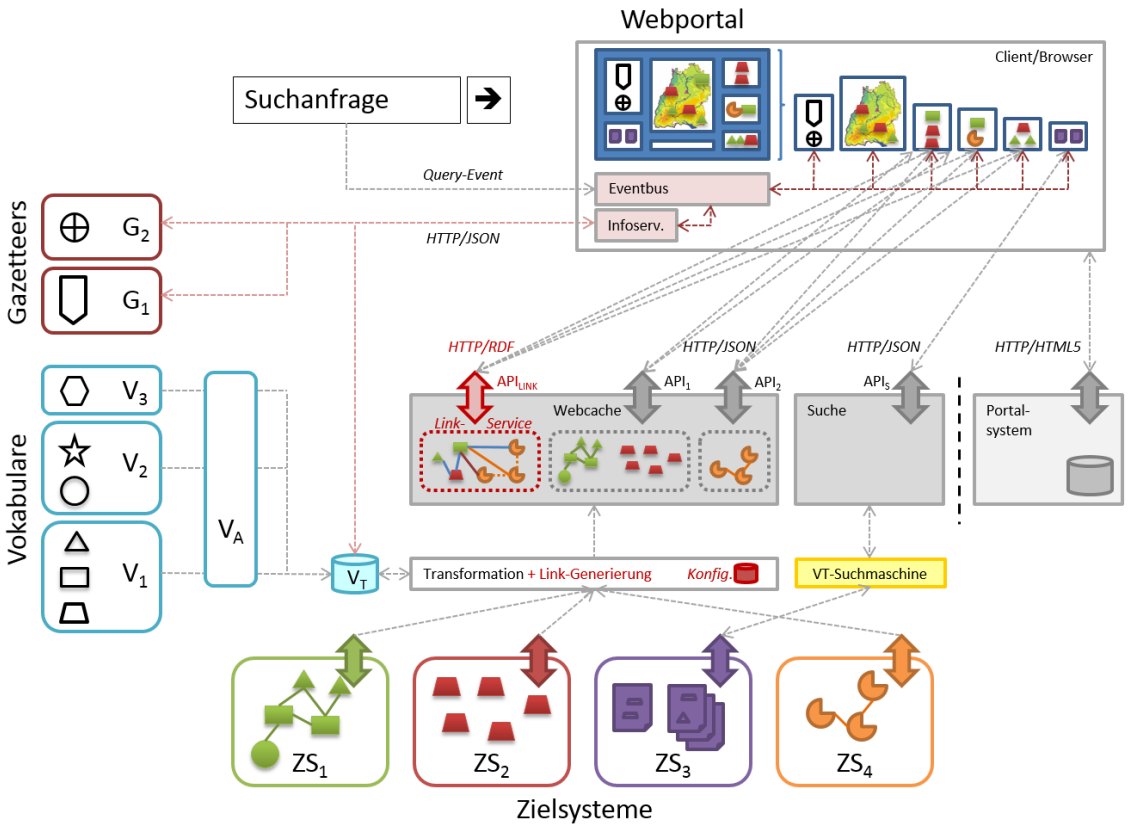

Abbildung 12: Umsetzung als serviceorientierte Architektur mit zusätzlichem Link-Service (neue Entwicklungen und eigene Anteile in rot)

Die Abbildung entspricht in ihrem Aufbau zunächst der dritten Architekturvariante. Hinzugekommen sind der Link-Service innerhalb des Webcache sowie die Link-Generierung (beide in rot), die parallel zur Transformation der Daten stattfindet. Die Transformationskonfiguration wird dafür dahingehend erweitert, dass z.B. zusätzliche Konfigurationsoptionen zur Generierung bzw. Verwendung eindeutiger IDs oder zur Beschreibung möglicher Relationen zwischen Klassen und Objekten hinterlegt werden

können. Die generierten Relationen können dabei entweder direkt aus den Originaldaten gewonnen werden, sofern die Datenlage es erlaubt (z.B. über Schlüssel-Fremdschlüssel-Beziehungen), oder zusätzlich nach Konfigurationsregeln bzw. durch Transformationsprozesse generiert werden.
Die Relationen werden durch eine weitere Schnittstelle (API_{LINK}) bereitgestellt. Sie bietet die Daten in standardisierten Varianten (RDF bzw. RDF/JSON) (W3C 2004b) an. Die Komponenten im Frontend (Webportal) können die Verknüpfungen zusätzlich zu den Daten abrufen und die Verknüpfungen je nach Anwendungsfall in ihren Darstellungen verwenden – die Darstellungen im Webportal oben rechts enthalten gegenüber der dritten Architekturvariante teilweise weitere Daten aus solchen Verknüpfungen.

Zu Implementierung des Link-Service stehen mehrere Varianten zur Verfügung. Für die Speicherung von RDF-Tripeln liegen zunächst sogenannte „subject-predicate-object databases“ (Revolvy 2016), die häufig auch „Triplestores“ genannt werden, nahe, z.B. Open-Source-Projekte wie Apache JENA (Apache Jena 2016) oder kommerzielle Datenbankerweiterungen zu IBM DB2 oder Oracle. Allerdings kann die Nutzung einer anderen Art von Backend-Software gegenüber den Triplestores Vorteile bringen, da Triplestores nur die einzelnen Tripel, nicht aber Zusammenhänge zwischen Tripeln bei identischen Subjekten bzw. Objekten speichern. Wenn die „Umgebung“ eines Objektes (über Relationen erreichbare Nachbarn) angefragt werden soll, d.h. eine Vielzahl über Relationen zusammenhängender Objekte, bietet sich die Nutzung einer Graphdatenbank an, die zunächst Objekte (ggf. inklusive ihrer gesamten Attribute) als Knoten, darüber hinaus jedoch auch die (attributierten) Verknüpfungen zwischen Objekten speichern kann. Je nach Anwendungsfall können so Daten beispielsweise sowohl (redundant) im Stammdatendienst als auch im Verknüpfungsdienst gespeichert, oder die Graphdatenbank auch als Backend-System für den Stammdatendienst genutzt werden - ggf. mit Performanznachteilen gegenüber indexierten Datenbank- oder Suchmaschinensystemen bei größeren Datenmengen. Einige Graphdatenbanken am Markt bieten Funktionalitäten und Vorgehensmodelle an, mit denen z.B. die Transformation ganzer Ontologien in eine Graphdatenbank möglich ist (Wiegand 2013), z.B. für die Graphdatenbank neo4j (neo4j 2016). Der Linkservice muss ggf. entsprechende Transformationen zwischen den Anfragen in SPARQL (W3C 2008) und die Anfragesprache bzw. das Antwortformat der Graphdatenbank vornehmen, sofern sie nicht ohnehin SPARQL als Abfragesprache unterstützt (DuCharme 2014).

3.6.1 Identität von Objekten

Grundlage für die Darstellung von Beziehungen zwischen Objekten ist, dass jedes Objekt über eine Identität verfügt, die meist durch einen eindeutigen Schlüssel ausgedrückt wird, der sich bei verschiedenen Objekten auch dann unterscheidet, wenn alle anderen Attribute gleiche Werte aufweisen.

In den meisten Quellsystemen werden Objekte bereits eine Identität besitzen. Dabei kann es sich um explizit vorgegebene Identitäten handeln (z.B. die eindeutige MAC-Adresse eines Netzwerkgerätes oder die eindeutig vergebene Nummer einer Messsta-

tion) oder um durch das technische System vergebene künstliche Identitäten, z.B. “Surrogate Keys”, d.h. fortlaufende Nummern in relationalen Datenbanken.

Sind in einem System keine expliziten Identitäten vorhanden, können sie in der Regel nach dem Muster von Primärschlüsseln relationaler Datenbanken gewonnen oder komplett künstlich erzeugt werden, z.B. einen Universally Unique Identifier, UUID.

Doch selbst wenn Objekte in einzelnen Quellsystemen bereits über Identitäten verfügen, müssen sie jedoch über Systemgrenzen hinweg nicht eindeutig sein, z.B. handelt es sich bei künstlichen Primärschlüsseln in relationalen Datenbanken meist um fortlaufende natürliche Zahlen, die Wahrscheinlichkeit der Mehrfachnutzung desselben numerischen Wertes ist bei mehreren relationalen Datenbanken daher sehr hoch. Um systemübergreifend eindeutige Identitäten zu schaffen, bietet es sich daher an, jedem Quellsystem eine Identität zu geben, z.B. eine einmalig künstlich generierte, eindeutige ID, und dann eine Verkettung der Quellsystem-ID mit der Objekt-Identität innerhalb des Quellsystems als systemübergreifend eindeutige Identität zu verwenden. Das ist nachfolgend veranschaulicht.

```
<uniqueObjectID> ::= <systemID>.<systemObjectID>
```

Die Identitäten aller Quellsysteme müssen daher an einer zentralen Stelle verwaltet werden, z.B. zusammen mit weiteren Metadaten der Quellsysteme.

3.6.2 Nutzung bzw. Generierung von Verknüpfungen

Beziehungen zwischen Objekten sind in den Quellsystemen in verschiedenen Varianten abgelegt. Sie können explizit oder implizit vorhanden sein und sich auf dasselbe oder auf ein anderes Quellsystem beziehen. Die wichtigsten davon werden nachfolgend zusammengestellt.

Explizite Beziehungen zwischen Objekten innerhalb eines Systems

Explizite Beziehungen zwischen zwei Objekten innerhalb eines Quellsystems bestehen dann, wenn beide Objekte in einem einzigen Quellsystem vorhanden und über eine Referenz oder mehreren Referenzen miteinander verbunden sind. Die technische Umsetzung der Referenzen hängt dabei vom Quellsystem ab und kann z.B. über Fremdschlüssel (relationale Datenbank), Zeiger (objektorientiertes System) oder Kanten (Graphdatenbank) realisiert sein.

Eine Beziehung zwischen zwei Objekten lässt sich in Form ihrer systemübergreifenden Identitäten

```
<systemID>.<systemObjectIDa> ~(Beziehungstyp)~ <sys-
temID>.<systemObjectIDb>
```

darstellen, wenn es sich dabei um eine ungerichtete Verknüpfung handelt, bzw.

```
<systemID>.<systemObjectIDa> -(Beziehungstyp)-> <sys-
temID>.<systemObjectIDb>
```

oder

```
<systemID>.<systemObjectIDa> <-(Beziehungstyp)- <sys-
temID>.<systemObjectIDb>
```

im Falle einer gerichteten Beziehung (beide Richtungen sind möglich) handelt.

Die SystemID ist dabei in beiden Fällen gleich und für die systemObjectID werden jeweils die Identitäten der Objekte verwendet.

Der Beziehungstyp erlaubt die Unterscheidung verschiedenartiger Beziehungen zwischen Objekten und entspricht dem Prädikat eines RDF-Tripels (W3C 2004c).

Alle expliziten Beziehungen zwischen Objekten innerhalb eines Systems lassen sich also bei Kenntnis der inneren Struktur des Systems automatisiert auf systemübergreifende Art darstellen.

Explizite Beziehungen zwischen Objekten über Systemgrenzen hinweg

Explizite Beziehungen zwischen zwei Objekten, die sich in verschiedenen Quellsystemen befinden, werden meist über die Nutzung von gemeinsamen (übergreifend festgelegten) Schlüsseln realisiert, z.B. die eindeutige Nummer einer Messstation.

Werden in beiden Systemen dieselben Schlüssel verwendet, lässt sich das als ungerichtete Beziehungen zwischen Objekten interpretieren, d.h. man kann von einem Objekt das andere erreichen und umgekehrt. Zum Beispiel können in einem System die Metadaten zu einer Messstation gespeichert sein, in einem zweiten System die zugehörigen Messdaten. Eine solche Verknüpfung ist ungerichtet und lässt sich über Systemgrenzen hinweg in der Form:

```
<systemIDa>.<systemObjectID> ~(Beziehungstyp)~ <sys-
temIDb>.<systemObjectID>
```

ausdrücken.

Eine zweite Variante der expliziten Beziehungen über Systemgrenzen hinweg ist die einseitige, gerichtete Beziehung eines Objekts a im Quellsystem A zu einem Objekt b im Quellsystem B. Dann ist, z.B. in einem Attribut von a, die Referenz (z.B. ID) auf das Objekt b im Quellsystem B gespeichert, das Objekt b besitzt jedoch seinerseits keine Referenz auf das Objekt a in A. Die gerichtete Beziehung von a zu b lässt sich folgendermaßen ausdrücken:

```
<systemIDA>.<systemObjectIDa> -(Beziehungstyp)-> <sys-
temIDB>.<systemObjectIDb>
```

Ein typisches Beispiel für eine gerichtete Beziehung ist die Zuordnung eines Objektes zu einer benannten Örtlichkeit (Lage des Objekts), z.B. Objekt a befindet sich im Landkreis b.

Explizite Beziehungen zwischen Objekten in verschiedenen Systemen lassen sich bei Kenntnis der inneren Struktur beider Systeme ebenfalls automatisiert darstellen, je nach Art der Verknüpfung in der gerichteten oder ungerichteten Form.

Implizite Beziehungen zwischen Objekten

Implizite Beziehungen zwischen zwei Objekten a und b (im Allgemeinen aus verschiedenen Quellsystemen A und B) bestehen dann, wenn aus den Eigenschaften von a über eine Abbildung (ggf. unter Hinzuziehung von Zusatzwissen) eine Beziehung zum Objekt b hergestellt werden kann. Die Abbildung kann operational außerhalb der Quellsysteme erfolgen, z.B. durch einen externen Dienst.

Stellen zum Beispiel alle Objekte aus A ihre geographische Position in Form von Geokoordinaten (z.B. als Latitute-Longitude-Paar) zur Verfügung und haben alle Objekte in B (z.B. Landkreise) eine flächenhafte Geometrie, so lässt sich über eine simple geometrische/geographische Operation bestimmen, in welchem Landkreis b ein Objekt a liegt. Sind die Landkreise flächendeckend verfügbar und haben paarweise disjunkte Flächen, ist die Abbildung eindeutig. Aus den Identitäten von a und b sowie deren Quellsysteme A und B lassen sich dann wieder explizite Beziehungen konstruieren, z.B. der Art

```
<systemIDA>.<systemObjectIDa> -("liegt in")-> <sys-
temIDB>.<systemObjectIDb>
```

wobei “liegt in” den (neuen) Beziehungstyp darstellt.

Implizite Beziehungen und lose Kopplung von Objekten

Geokoordinaten sind ein gutes Beispiel für die weit reichenden Möglichkeiten von impliziten Beziehungen zwischen Objekten. Da in der realen Welt sehr viele Objekte eine explizite Geometrie (z.B. Gemeinden, Landkreise, Schutzgebiete) bzw. einen Standort haben (z.B. Messstationen, Umspannwerke, Hochspannungsmasten), die häufig auch in den entsprechenden Objektdaten verfügbar sind, lassen sich Beziehungen zwischen Objekten auf rein geometrischer/geographischer Ebene herstellen (z.B. „ist enthalten in”, „schneidet”, „ist in der Nähe von”). Die Geokoordinaten können z.B. auch ohne das explizite Herstellen von Beziehungen genutzt werden, z.B. können alle Objekte aus einen Quellsystem A ermittelt werden, die innerhalb einer bestimmten Fläche oder im Umkreis von 2 km um einen bestimmten Punkt liegen.

Sind Objekte bezüglich ihrer (Geo-)Attribute abfragbar, lassen sich Beziehungen direkt in der konsumierenden Anwendung verwenden, z.B. um alle erreichbaren Objekte in einem gegebenen Kartenausschnitt anzuzeigen.

Einen weiteren Kandidaten für lose Kopplungen stellen zeitliche Bezüge (Zeitpunkte, Zeiträume) dar.

Indirekte Beziehungen zwischen Objekten und Ableitung neuer Beziehungen

Gegebenenfalls lassen sich aus bestehenden Beziehungen weitere Beziehungen konstruieren bzw. ableiten, insbesondere wenn mittelbare Beziehungen bestehen. Hat ein Objekt a eine Beziehung zu Objekt b und Objekt b eine Beziehung zu Objekt c, so lässt sich daraus eine Beziehung zwischen den Objekten a und c konstruieren (Transitivität).

Zum Beispiel kann ein Objekt a im Landkreis b liegen und der Landkreis b Teil des Bundeslandes c sein. Dann lässt sich eine Beziehung „Objekt a liegt in Bundesland c“ bilden. Zur Konstruktion solcher Beziehungen bedarf es jedoch weiterer Informationen bzw. des entsprechenden Fachwissens, das z.B. durch eine formale Semantik oder durch Inferenzdienste bereitgestellt werden kann, so dass Ableitungen ggf. auch automatisiert gewonnen werden können (Reasoning).

3.6.3 Beziehungsdienst

Wenn Daten aus Quellsystemen über einheitliche (Micro-)Services mit RESTful URLs bereitgestellt werden, sollen Objektidentitäten und Beziehungen zwischen Objekten selbstverständlich erhalten bleiben. Dazu wird der Beziehungsdienst benötigt.

Auf der Seite der Quellsysteme werden Identitäten von Objekten nach dem Vorschlag aus Abschnitt 4.5.1 anhand von Schlüsseln nach dem Muster

```
<uniqueObjectID> ::= <systemID>.<systemObjectID>
```

eindeutig identifiziert.

Die RESTful URLs der einzelnen Dienste haben eine Form nach dem Muster

```
https://<baseurl>/<function>/<domain>/<type>/<id>/<aspect>
```

Die `<baseurl>` gibt dabei eine reale Server Adresse an, über die alle Anfragen bedient werden (z.B. ein Gateway bzw. ein Dispatcher-Dienst).

`<function>` dient der Unterscheidung von (generischen) Grundfunktionalitäten bzw. -diensten, z.B. zur Bereitstellung von Objektdaten, Zeitreihen oder Mediendaten.

`<domain>` kann der weiteren Strukturierung von Daten dienen, z.B. wenn gleichartige Daten für verschiedene Bundesländer vorliegen und danach unterschieden werden sollen. Die Angabe einer Domain ist optional, eine `<domain>` kann aber ggf. auch mehrere Hierarchiestufen einnehmen, z.B. "`/bw/ka`" oder (intern) "`bw.ka`" wenn innerhalb eines Bundeslandes auch nach Regierungspräsidien unterschieden werden soll.

`<type>` dient der inhaltlichen Unterscheidung von Objekten, zum Beispiel nach deren Klasse/Konzept und ist zwingend notwendig.

Die `<id>` bezeichnet ein konkretes Objekt des Typs. Ist keine `<id>` angegeben, so werden alle Objekte des Typs geliefert bzw. alle die zu den weiteren Parametern passen.

Der optionale Bestandteil `<aspect>` kann der weiteren Diskriminierung der Anfrage dienen.

Er kann durch weitere Parameter ergänzt werden.

Konkret liefert z.B. die URL

```
https://linked-energy.org/objects/bw/windturbines/303
```

die Objektdaten zu einer bestimmten Windkraftanlage in Baden-Württemberg, während die URL

```
https://linked-
energy.org/measures/bw/windturbines/303/power
```

die zugehörigen Messwerte (z.B. die aktuelle Einspeiseleistung) liefert.

Um eine eindeutige Zuordnung von Daten aus den Quellsystemen und den über die RESTful APIs bereitgestellten Repräsentationen gewährleisten zu können, z.B. zur Synchronisation der Daten bei Änderungen auf der einen oder der anderen Seite oder zur Auflösung von Beziehungen, besteht grundsätzlich die Notwendigkeit einer Abbildung der eindeutigen, aus den Quellsystemen konstruierten `<uniqueObjectID>`, die zur Adressierung der Daten im Quellsystem benötigt wird und der entsprechenden RESTful URL, welche die Daten im (Micro-)Service adressiert. Die Abbildung muss in beiden Richtungen funktionieren.

Mithilfe einer solchen Abbildung ist es zum Beispiel möglich, bei der Synchronisierung von Daten aus dem Quellsystem mit einem Service die Beziehungen zu anderen Objekten zunächst durch Verwendung der eindeutigen `<uniqueObjectID>`s und anschließend durch deren Substitution durch RESTful URLs die Anforderung 4 zu Linked Data (s. Abschnitt 3.6) zu erfüllen.

Der Beziehungsdienst muss dazu Funktionen in beiden Abbildungsrichtungen bereitstellen, z.B. nach den URL-Mustern

```
https://linked-energy.org/links/uniqueid/<urlFragment>
```

oder

```
https://linked-energy.org/links/uniqueid?url=<encodedUrl>
```

bzw.

```
https://linked-energy.org/links/url/<uniqueId>
```

oder

```
https://linked-energy.org/links/url?id=<uniqueId>
```

Die notwendigen Daten für die Abbildung können beim Anschluss eines Quellsystems gewonnen werden. Es muss eine eindeutige ID für das Quellsystem vergeben und das Schlüsselattribut (ggf. künstlich) für die Objekte innerhalb des Quellsystems festgelegt werden. Bei der Definition des URL-Raums des zugehörigen Dienstes bzw. der zugehörigen Dienste müssen ggf. die mögliche(n) Funktion(en), die Domain sowie der Typ der Daten festgelegt werden.

Damit stehen alle notwendigen Informationen für die Abbildung zur Verfügung und bei der Synchronisation der Daten können dem Beziehungsdienst sowohl `uniqueObjectID`s als auch URLs zur Verfügung gestellt werden.

In vielen Fällen lässt sich der Aufwand durch die Nutzung von Konventionen deutlich reduzieren, dann lassen sich `uniqueObjectID`s und URLs auf rein syntaktischer Ebene ineinander überführen, was im Folgenden dargestellt wird.

Konvention für die Abbildung von Objektschlüsseln auf RESTful URLs

In einem konkreten Beispiel sind die Daten aller Windkraftanlagen aus Baden-Württemberg in einem einzigen Quellsystem verfügbar und mit eindeutigen nummerischen IDs versehen.

Wählt man den Schlüssel für das Quellsystem mit Bedacht, z.B. den Namen eines Konzeptes, das dem Inhalt der Datenquelle entspricht, wie “windturbine”, und verwendet zur Identifikation der einzelnen Objekte deren Schlüssel im Quellsystem, dann entstehen eindeutige Schlüssel für das Quellsystem nach dem Muster

```
<uniqueObjectID> ::= <systemID>.<systemObjectID>
```

für die Windturbine mit der ID 303 in der konkreten Form

```
windturbine.303
```

Für den in der Praxis durchaus häufigen Fall, dass gleichartige Objekte aus genau einem Quellsystem stammen, lässt sich mit Hilfe der vorgestellten Konvention erreichen, dass eine Abbildung von RESTful-URLs und den `uniqueObjectID`s zur eindeutigen Identifikation von Objekten trivial, d.h. eine rein syntaktische Umwandlung, ist. Ggf. müssen dazu Metadaten des Quellsystems genutzt werden, z.B. die zugehörige Domäne (`<domain>`).

Abbildungen zwischen `uniqueObjectID`s und URLs können so leicht berechnet werden, d.h. sie müssen nicht explizit gespeichert werden.

3.6.4 Metadatendienst

Metadaten sollen die Inhalte eines Informationssystems und ggf. die Möglichkeiten zum Zugriff darauf genauer beschreiben. Sie sollen die Nutzung der Daten durch andere Systeme ermöglichen, d.h. sie liefern einen wichtigen Beitrag zur Interoperabilität von Systemen. Daher beinhalten Metadaten Informationen zur Semantik, zum Datenmodell und zur Syntax der Daten.

Im Sinne der vorgestellten Architektur kann es Metadaten auf zwei Ebenen geben, der Ebene der Quellsysteme sowie der bereitgestellten (Micro-)Services.

Erstere sollten eigentlich zu jedem Quellsystem existieren, in der Realität liegen jedoch häufig keine Metadaten vor, zumindest nicht in einer standardisierten bzw. maschinell verarbeitbaren Form.

Für alle (Micro-)Services bzw. für alle durch sie verfügbar gemachten Datenquellen sollen ebenfalls Metadaten bereitgestellt werden. Zwei Varianten bieten dabei unterschiedliche Sichten auf die Systeme - die öffentlich verfügbaren Metadaten, welche die oben beschriebenen Informationen zum Zugriff (Semantik, Datenmodell und Syntax) auf den Service enthalten, sowie eine interne Sicht, die darüber hinaus weitere Informationen, z.B. über die verwendeten Schlüsselattribute im Quellsystem, den eindeutigen Schlüssel für des zugehörigen Quellsystems und weitere für die Datensynchronisation notwendige Angaben (z.B. Abbildungsvorschriften für Attribute, Aktualisierungsintervalle etc.) enthält.

Der Metadatendienst dient in der zweiten Sicht somit auch der Verwaltung der angeschlossenen Quellsysteme.

3.7 Gegenüberstellung der vier Architekturvarianten

Die vier in den vorigen Abschnitten vorgestellten Architekturvarianten bauen aufeinander auf bzw. in den späteren Varianten wurden die Schwächen ihrer Vorgänger in der Architektur beseitigt.

Die folgende Tabelle stellt die vier Architekturvarianten kompakt gegenüber und zeigt ihre Vor- und Nachteile auf:

Architekturvariante mit wesentlichen Merkmalen	**Vorteile**	**Nachteile**
Erste Architekturvariante: Semantische Erweiterung einer kommerziellen Volltextsuchmaschine, Nutzung externer Datenquellen über OneBox-Mechanismus	Schnelle Umsetzung auf Basis eines vorhandenen Portals und einer Suchmaschine, seit über 10 Jahren produktiv Umweltthesaurus leicht in die Volltextsuche integrierbar Schnelle Anbindung neuer Zielsysteme über OneBox-Adapter Erweiterung der Suchanfrage lässt sich ergänzend, aber auch unabhängig von anderen Suchmechanismen nutzen	Keine echte Nutzung von Semantik OneBoxes profitieren nicht vom Umweltthesaurus Einschränkungen bei der Ergebnispräsentation (OneBoxes nicht in Trefferliste integriert) Geringe Bandbreite von Datentypen (keine Karten, Zeitreihen etc.) durch Beschränkung auf Volltextsuche Fehlende Standardisierung: Zur Anbindung jedes Zielsystems als OneBox ist ein Adapter notwendig
Zweite Architekturvariante: Serverseitige Verarbeitung der Suchanfrage, SearchBroker und Ontologiesystem	Mächtige Vorverarbeitung und semantische Verarbeitung der Suchanfrage Flexibles Ontologiesystem mit Artikulationsontologie zur Abbildung von Vokabularen und Harmonisierung heterogener Informationssysteme Technische und semantische Beschreibung von Zielsystemen	Serverseitiger Flaschenhals Wenig flexible Ergebnispräsentation, da serverseitig generiert Wenig Interaktion für den Nutzer (Weiternavigation) Große Anzahl von Schnittstellen im Suchsystem zur Anbindung vieler heterogener Zielsysteme Ontologiesystem verkraftet keine großen Zahlen von

	Größere Bandbreite von Datentypen (Tabellen, Karten)	Individuen Manuelle Arbeit bei der Harmonisierung von Vokabularen/Ontologien
Dritte Architekturvariante: Serviceorientierung, „Webcache“, clientseitige Verarbeitung	Flexible und leistungsfähige Sammlung von Diensten (SOA) („Webcache“) Modularer, Komponentenbasierter Client mit rein clientseitiger Kommunikation (EventBus) und vielen Darstellungsvarianten Asynchrone Verarbeitung von Anfragen an die Services Umfangreiche Unterstützung aller in den Domänen „Umwelt“ und „Energie“ benötigten Datentypen	Nicht in das Semantic Web integriert, kein Linked Data Semantische Hintergrundinformationen werden fast nur beim Transformationsprozess (Data Ingestion) genutzt Große Anzahl von Schnittstellen beim Transformationsprozess (Data Ingestion) zur Anbindung vieler heterogener Zielsysteme Manuelle Arbeit bei der Harmonisierung von Vokabularen
Vierte Architekturvariante: Ausbau zu semantischen Diensten, Linked Data	Unterstützung von Linked Data Generierung zusätzlichen Wissens (z.B. Beziehungen) gegenüber den originalen Zielsystemen Einbeziehung globaler Schemata, z.B. schema.org, dadurch Erhöhung möglicher Interoperabilität	Noch nicht vollständig umgesetzt Große Anzahl von Schnittstellen beim Transformationsprozess (Data Ingestion) zur Anbindung vieler heterogener Zielsysteme Manuelle Arbeit bei der Harmonisierung von Vokabularen

Tabelle 1: Die Vor und Nachteile der vier Architekturvarianten gegenübergestellt.

Bei der vierten Architekturvariante bleiben am Ende keine wesentlichen inhaltlichen Nachteile stehen, außer denen, die sich, wie auch bei den vorhergehenden Varianten, direkt aus der Heterogenität der betrachteten Zielsysteme ergeben. Die vierte Architekturvariante ist jedoch noch nicht vollständig, d.h. etwa zu 90%, umgesetzt.
Im Gegensatz dazu gibt es für die ersten drei Architekturvarianten zahlreiche praktische Umsetzungsbeispiele, die im folgenden Kapitel 4 beschrieben werden.

4 Umsetzungsbeispiele

Die folgenden Umsetzungsbeispiele demonstrieren die verschiedenen Architekturvarianten in realen Systemen. Dabei wird zunächst gezeigt, wie Architekturvariante 1 (Semantische Erweiterung von Suchanfragen und Nutzung externer Datenquellen durch die Volltextsuchmaschine) die Suche im Energieportal Baden-Württemberg verbessert hat. Die Nutzung von Architekturvariante 2 (Serverseitige Verarbeitung der Suchanfrage, SearchBroker und Ontologiesystem) wird anhand des Projektes SUI (Semantische Suche nach Umweltinformationen) demonstriert. Die dritte Architekturvariante (Serviceorientierung, „Webcache“, clientseitige Verarbeitung) wird anhand des Energieatlas Baden-Württemberg, der Landesumweltportale (LUPO) sowie der Nutzung durch mobile Apps demonstriert. Für die vierte Architekturvariante (Ausbau zu semantischen Diensten / Linked Data) liegen bisher noch keine produktiven Umsetzungsbeispiele vor.

4.1 Energieportal Baden-Württemberg

Nachdem im Jahr 2011 in Baden-Württemberg die Themen Umwelt, Klima und Energiewirtschaft in einem Ministerium gebündelt worden waren, entstand der Bedarf nach einem Energieportal Baden-Württemberg, das einen zentralen Einstiegspunkt zu behördlichen Informationen zum Thema Energie darstellen sollte. Zielgruppe des Portals war die allgemeine Öffentlichkeit. Ein solches Portal konnte mit Hilfe des existierenden LUPO-Baukastens (Schlachter et al. 2011b), d.h. einer Sammlung von Diensten, Komponenten und Technologien, die auch in den (damaligen) Landesumweltportalen von Baden-Württemberg[17], Sachsen-Anhalt[18] und Thüringen[19] eingesetzt werden, aufgesetzt und mit Inhalten gefüllt werden, so dass es zur CeBit 2012 der Öffentlichkeit präsentiert werden konnte.

Abbildung 13 zeigt die Hauptthemen und inhaltlichen Zugänge des Energieportals Baden-Württemberg. Für den schnellen Zugang zu aktuell besonders gefragten Themen werden entsprechende Begriffe in Form einer Tagcloud (Abbildung 13) dargestellt. Ein Klick auf den entsprechenden Begriff triggert eine Volltextsuche nach dem entsprechenden Thema. Bei den in der Tagcloud hinterlegten (redaktionell gepflegten) Begriffen handelt es sich in der Regel um Begriffe, die im Umweltthesaurus enthalten und mit Synonymketten versehen sind, so dass sie bei der Verwendung als Suchbegriffe direkt von der automatischen Suchworterweiterung profitieren.

[17] https://www.umwelt-bw.de

[18] https://www.umwelt.sachsen-anhalt.de

[19] http://www.umweltportal.thueringen.de

Abbildung 13: Menüstruktur (oben) und Tagcloud (unten) im Energieportal Baden-Württemberg (Screenshots)

Das Energieportal Baden-Württemberg wurde im Auftrag des Ministeriums für Umwelt, Klima und Energiewirtschaft technisch vom Institut für Angewandte Informatik (IAI) und inhaltlich von der Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg (LUBW) umgesetzt. Beim Aufbau des Energieportals haben sich die bereits verfügbaren LUPO-Komponenten, die allesamt vom Autor der vorliegenden Arbeit entwickelt wurden, bewährt. Für die Volltextsuche im Energieportal wurde die im Rahmen der vorliegenden Arbeit erstellte Komponente zur automatischen Suchworterweiterung erstmals produktiv eingesetzt. Die vorgeschlagenen Suchbegriffe liefert ein Energie-Wörterbuch, das auf Grundlage der deutschen Ausgabe des GEMET-Thesaurus (European Environment Information and Observation Network 2017) neu aufgebaut wurde. Dem Nutzer werden nun bereits bei der Eingabe einiger Zeichen im Suchschlitz passende Suchbegriffe aus dem Energiebereich angeboten. Synonymketten aus dem GEMET wurden als weiteres Wörterbuch in der Suchmaschine hinterlegt, was insbesondere bei der Verwendung umgangssprachlicher Begriffe in der Suchanfrage hilft, da sie auf die entsprechenden Fachbegriffe abgebildet und bei der Suche verwendet werden. Für die Umweltportale steht ebenfalls ein umfangreicheres Wörterbuch auf derselben Basis (GEMET) zur Verfügung, das alle Umweltbereiche abdeckt.

Im Energieportal Baden-Württemberg wurden erstmals Kartendarstellungen in die LUPO-Trefferlisten der Volltextsuche integriert, zunächst für Karten aus den Bereichen „Solare Effizienz auf Hausdächern“, „Windenergie“ und „Wasserkraftpotenziale am Neckar“ (Abbildung 14). Die Darstellung der Karten erfolgt in einer vom übrigen Layout

abgesetzten FancyBox (fancybox.net 2016). Mit der Möglichkeit zur erweiterten Parametrisierung von Aufrufen des Umweltdaten- und Kartendienstes (UDO) der Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg (LUBW) ist ein Orts- bzw. Adress-scharfer Einsprung in die Kartenansichten möglich.

Eine wichtige Synergie ergibt sich aus der starken inhaltlichen Überschneidung des Themas „Energie“ im Umweltportal mit den Inhalten des Energieportals. Hierfür wurde ein Konzept entwickelt, das die gemeinsame Nutzung des Volltextindexes in beiden Portalen vorsieht und dabei den Pflegeaufwand auf ein Mindestmaß reduziert.

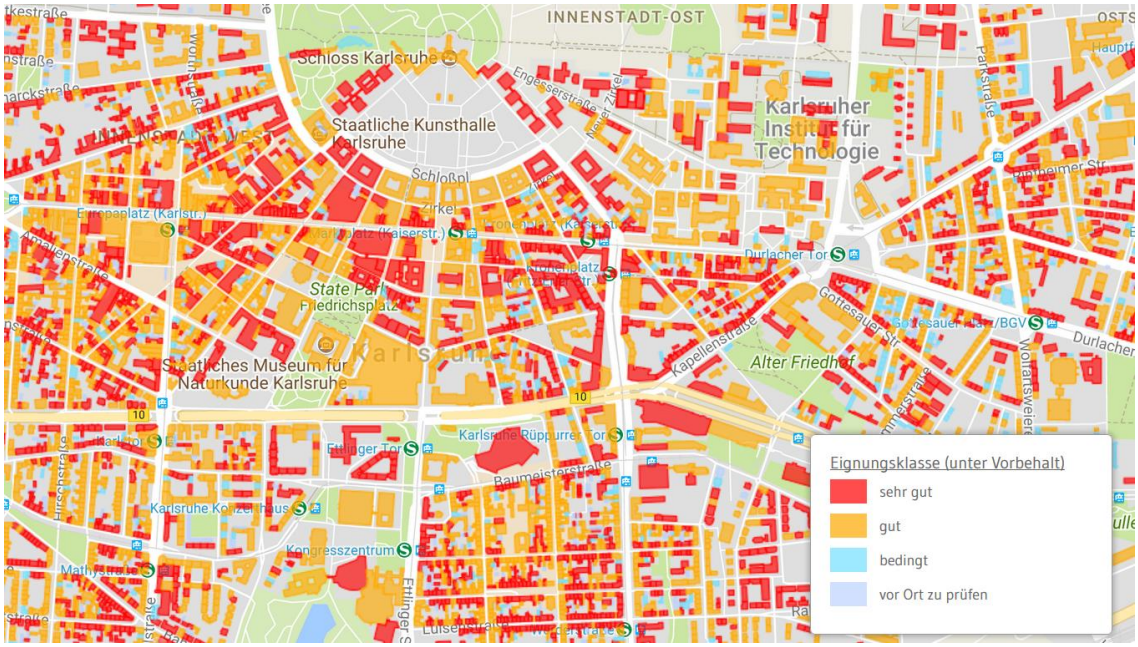

Abbildung 14: Beispiel für Kartendarstellung im Energieportal Baden-Württemberg: Eignung von Dachflächen für Photovoltaikanlagen auf Basis der solaren Einstrahlung (Screenshot). Dieselbe Darstellung wird auch im Energieatlas Baden-Württemberg verwendet.

Mit der Inbetriebnahme des Energieatlas Baden-Württemberg (Abschnitt 4.3) wurde das Energieportal außer Betrieb genommen.

4.2 Semantische Suche nach Umweltinformationen (SUI)

Im Projekt „Semantische Suche nach Umweltinformationen“ (SUI) des Fraunhofer Instituts IOSB und des Instituts für Angewandte Informatik (IAI) am KIT wurden in Zusammenarbeit mit der LUBW und dem UM Baden-Württemberg Informationstechnologien entwickelt und evaluiert, mit denen die Volltextsuche in Umweltportalen durch den Einsatz von semantischen Technologien und Serviceorientierung verbessert werden kann.

Das Konzept von SUI basiert auf der Architekturvariante 2 (Serverseitige Verarbeitung der Suchanfrage, SearchBroker und Ontologiesystem), d.h. darauf, dass die Semantik der Informationen von ausgewählten Fachinformationssystemen (sogenannter Zielsysteme) über Zielsystembeschreibungen, die eine Kategorisierung des Inhalts des Ziel-

systems gemäß spezifischer Informationsklassen, Orts- und/oder Zeitbezug vornehmen, im Portal zusammen mit Informationen zum Abruf der Informationen im Zielsystem gespeichert wird. Im Rahmen einer intelligenten Vorverarbeitung von Suchbegriffen im Portal, die zur inhaltlichen Klassifikation der Suchbegriffe die im Rahmen des SUI-Projektes entwickelte SUI-Ontologie verwendet, werden Suchbegriffe dann auf die jeweiligen inhaltlichen Kategorien, Orts- und Zeitangaben abgebildet. Anschließend werden passende Suchergebnisse über die Serviceschnittstellen ermittelt, d.h. aus den ausgewählten Zielsystemen extrahiert und schließlich im Portal in einer Suchergebnisseite übersichtlich angezeigt.

Die Gesamtarchitektur von SUI entstand größtenteils als Beitrag der vorliegenden Arbeit. Die Szenarien für SUI wurden gemeinsam durch IAI, IOSB und LUBW entwickelt. Die serverseitige Informationsverarbeitung (Search Broker, Vorverarbeitung, Abfrage des Ontologiesystems, Zielsystembeschreibungen, Abfrage der Zielsysteme) wurden durch den Autor der vorliegenden Arbeit konzipiert und implementiert, ebenso die Präsentation der Ergebnisse im Webgenesis-basierten Portalsystem. Das Ontologiesystem wurde in Form eines weiteren Webgenesis-Systems unter Einbeziehungen von Ergebnissen des THESEUS-Projektes (Bundesministerium für Wirtschaft und Energie 2017) durch das IOSB bereitgestellt, ebenso die Oberfläche für das Ontologie-Mapping. Die Verwendung der Teilontologien, das Konzept für das Ontologie-Mapping und der Algorithmus zur Abfrage der Ontologie wurde unter Mitwirkung des Autors gemeinsam durch das IOSB und das IAI konzipiert.

In einem ersten Schritt zur Spezifikation geeigneter Serviceschnittstellen werden Inhaltsangebote und Zugriffsschnittstellen ausgewählter Zielsysteme analysiert und darauf aufbauend wird eine Spezifikation zur formalen Beschreibung von Zielsystemen auf Basis des OpenSearch-Description-Standards (opensearch.org 2013) entwickelt. Hier wurden dann vor allem CMS-basierte Zielsysteme, wie der Themenpark Umwelt (Düpmeier et al. 2007; Düpmeier et al. 2008; Düpmeier et al. 2009) und das Fachdokumentenmanagementsystem FADO (Weidemann et al. 2007; Weidemann et al. 2008; Weidemann et al. 2009; Weidemann et al. 2010) als Zielsysteme in die SUI-Suche eingebunden. Dazu wurde für den Themenpark Umwelt zunächst eine SUI-kompatible Serviceschnittstelle entworfen und implementiert, die schließlich durch flexible Konfigurationsmechanismen so generalisiert wurde, dass sie sich auf beliebige Webgenesis-basierte Systeme, wie FADO, übertragen ließ.

Der Themenpark Umwelt eignete sich deshalb gut als erstes Entwicklungssystem für die CMS Service API, da die verschiedenen Arten und Typen von Informationen im Themenpark in Bezug auf verschiedene Inhalts- und Medienklassen typisiert und stark strukturiert sind. Weiter gibt es im Themenpark Inhaltsobjekte mit räumlichem Bezug. Daher lassen sich alle Facetten der spezifizierten Service-API auch durch Themenpark-Inhalte abbilden und testen (Abecker et al. 2011).

Ontologien und Ontologiemapping

Da die Neuentwicklung von Ontologien teuer und zeitaufwändig ist, wird in SUI die Gewinnung von Ontologien aus vorhandenen Quellen favorisiert (Bügel et al. 2011b). Gerade im Bereich Umweltinformation ist eine beträchtliche Anzahl terminologischer Systeme verfügbar, in deren Entwicklung bereits viel Aufwand investiert wurde. Ein großes Spektrum der in SUI benötigten thematischen Modellierung ist mit der Verfügbarkeit der Systeme bereits abgedeckt. In SUI wird daher der Ansatz verfolgt, Systeme direkt zu nutzen und den Schwerpunkt auf die Entwicklung von Werkzeugen für deren (semi-)automatische Konvertierung in die gewünschte Repräsentationsform zu legen. Dazu gehören vor allem die im Rahmen der Semantic Web Initiative standardisierten Formate OWL, RDF oder SKOS. Der Vorteil ihrer Nutzung besteht vor allem in der Möglichkeit zur Nutzung eines großen Repertoires an Verarbeitungswerkzeugen, die im Kontext des Semantic Web entstehen bzw. bereits heute verfügbar sind. Ein weiterer Vorteil ergibt sich aus der Minimierung des Aufwandes für die Pflege der Ontologien. Die Pflege kann prinzipiell an der Quelle erfolgen, d.h. bei einer Weiterentwicklung der originalen Systeme kann durch einen definierten Updatevorgang auch die erzeugte Ontologie auf den neuesten Stand gebracht werden.

Im SUI-System werden zurzeit die folgenden Ontologien eingesetzt:

- GEMET (GEneral Multilingual Environmental Thesaurus): Der Thesaurus ist im SKOS-Format verfügbar und bietet eine umfassende Basis von Suchtermen. Er ist auch Bestandteil des Semantic-Network-Service (SNS) des Umweltbundesamtes (Umweltbundesamt 2016).
- Objektarten-Katalog (OK): In integrierten Umweltinformationssystemen und Geodatenverbünden werden für die Fachbereiche alle Objektarten, die Realweltphänomene repräsentieren, in sog. Objektarten-Katalogen (bei ISO/OGC auch Feature Type Catalogues genannt) geführt und damit der Datenaustausch erleichtert. Wenngleich nicht ursprünglich für den Einsatz in Suchportalen konzipiert, liefert der OK wichtige Informationen für den Zugriff auf Umweltinformationen in Datenbanken, z.B. Objektartencodes und Fachführungscodes. Im Rahmen von SUI wurde der OK in eine Ontologie transformiert.
- Lebenslagen-Ontologie: Verschiedene Landesportale, z.B. das vom Innenministerium Baden-Württemberg betriebene Portal „service-bw“ (Service-BW 2016), bieten bereits eine Navigation durch ihre Webseiten auf Basis definierter Lebenslagen an. Im Rahmen von SUI wurde die Lebenslagen-Struktur von service-bw prototypisch mit Umweltthemen ergänzt.

Die Liste eingesetzter Ontologien ist offen, d.h., es gibt Pläne zur Integration weiterer verfügbarer Systeme. Die einzelnen Ontologien müssen jedoch miteinander in Verbindung gebracht werden, was im Folgenden beschrieben wird (Abbildung 15):

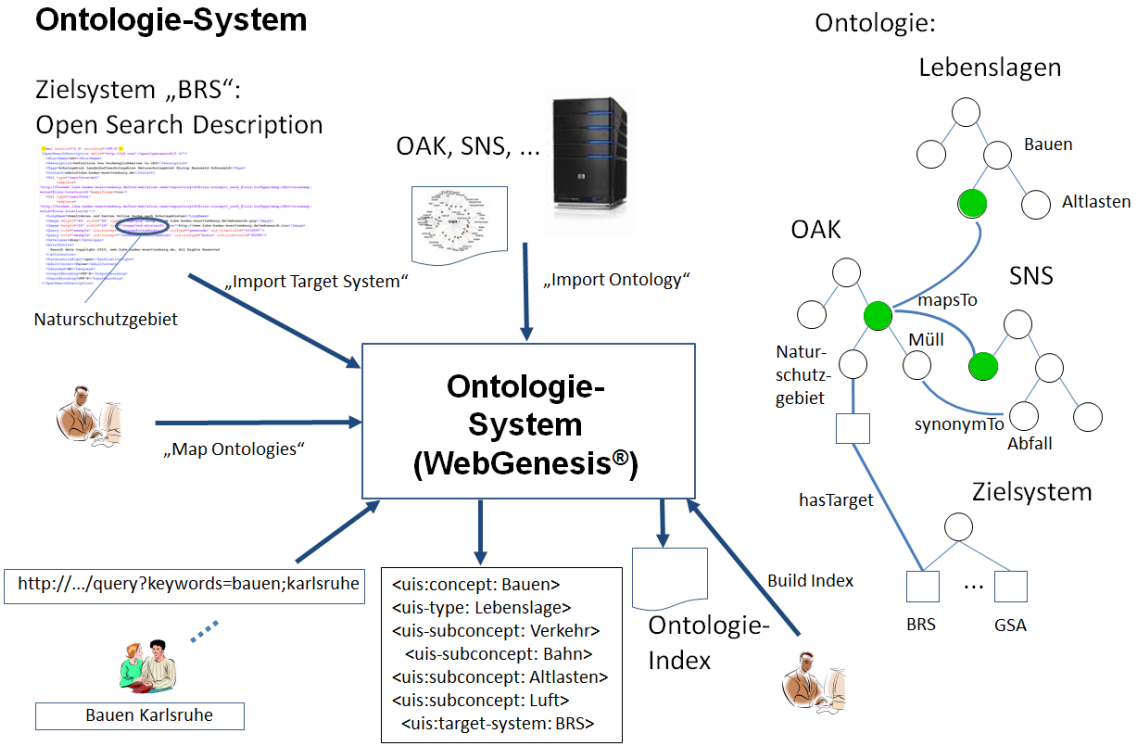

Abbildung 15: Ontologiesystem in SUI; aus: (Bügel et al. 2011b)

Die eingesetzten Ontologien (bzw. die zugrundeliegenden originalen Begriffssysteme) werden unabhängig voneinander durch unterschiedliche Experten-Gremien entwickelt, z.B. Lebenslagen, Objektartenkatalog (OAK) oder der Umweltthesaurus (SNS) auf der rechten Seite. Sie modellieren spezifisches Wissen, das auf die jeweilige Domäne fokussiert ist. Querbezüge zwischen Domänen werden hierdurch nicht erfasst. Prinzipiell können identische Phänomene in unterschiedlichen Domänen durch unterschiedliche Begriffsstrukturen und Relationen modelliert sein. Das SUI-System muss daher in der Lage sein, Querbezüge zur Nutzung durch die semantische Suche explizit zu machen.

Ontologien müssen daher begrifflich aufeinander abgebildet werden. Die im Kontext einer Suchanfrage vom Ontologiesystem automatisch berechnete thematische Erweiterung der Anfrage muss sich über verschiedene Ontologien erstrecken. Liefert beispielsweise eine Anfrage nach „Bauland“ einen Treffer in GEMET, ermittelt das Ontologiesystem auch die semantische Umgebung des Begriffs „Bauland“ ausschließlich aus Begriffen aus GEMET. Es liefert jedoch nicht die Lebenslage „Bauen“, die in der Lebenslagen-Ontologie modelliert ist.

Im SUI-System kann nun durch Herstellung einer expliziten Beziehung zwischen beiden Begriffen das Ergebnis der thematischen Aufbereitung entscheidend verbessert werden.

Für die Erzeugung der Querbezüge steht eine breite Palette automatisch arbeitender Werkzeuge für das Ontologie-Mapping zur Verfügung. In SUI wird ein Werkzeug eingesetzt, das im Rahmen des deutschen Forschungsprogramms THESEUS (Bundesministerium für Wirtschaft und Energie 2017) u.a. durch das Fraunhofer-Institut für Opt-

ronik, Systemtechnik und Bildauswertung (IOSB) entwickelt wurde und speziell für große Ontologien ausgezeichnet skaliert. Werkzeuge für automatisches Ontologie-Mapping sind meist in Form einfacher Konsolen-Programme verfügbar (Euzenat und Shvaiko 2007). Die Integration in eine konkrete Anwendung erfordert Ergänzungen hinsichtlich der Nutzbarkeit und Anwenderfreundlichkeit.

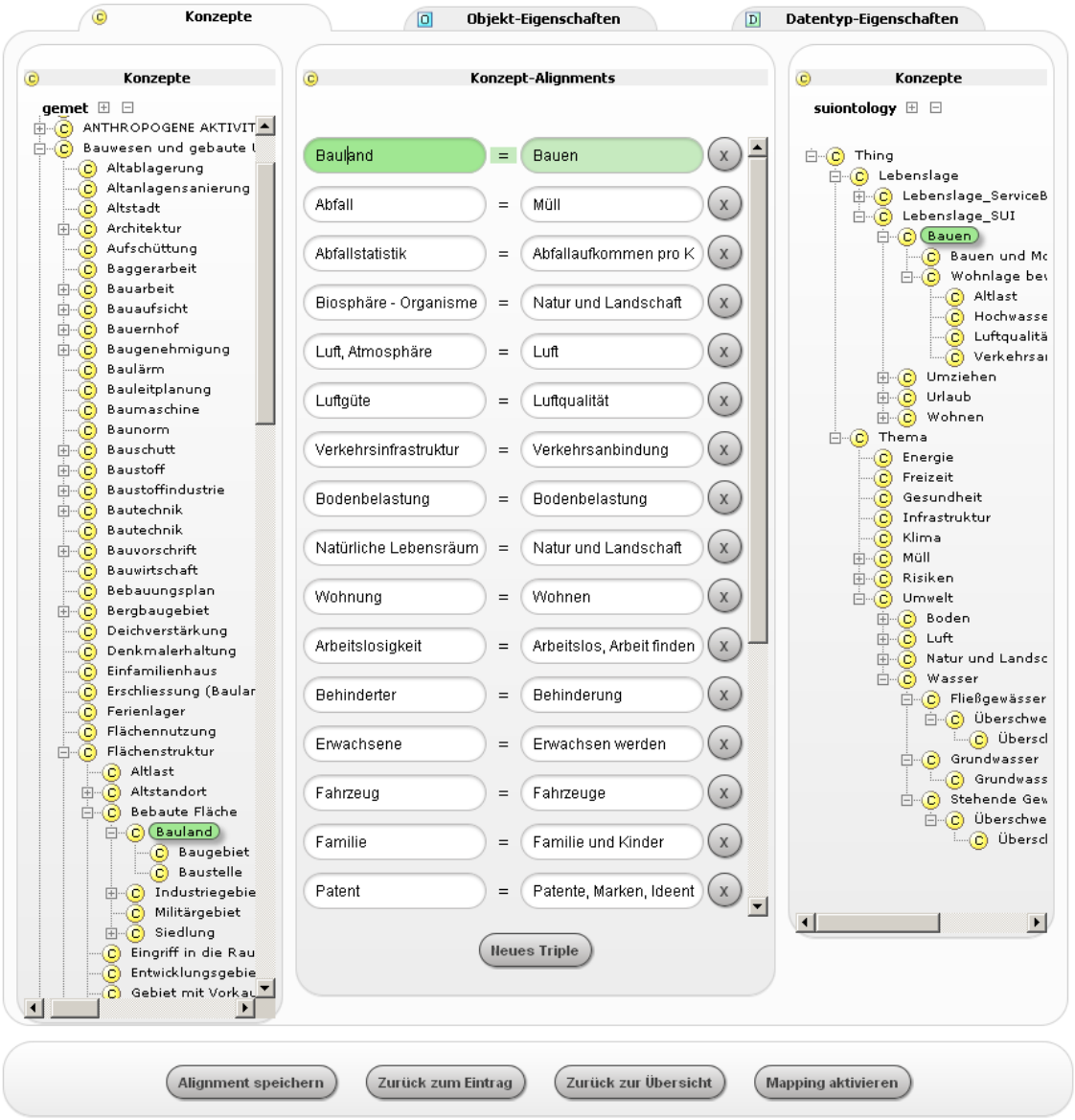

Abbildung 16: Ontology-Mapping im SUI-System; aus: (Bügel et al. 2011b)

Da Mappings zwischen Ontologien bilateral sind, andererseits aber in der Anwendung beliebig viele Ontologien zum Einsatz kommen können, wurde für SUI eine einfach handhabbare, Web-basierte Verwaltung automatisch erzeugter Mappings entwickelt und in das SUI-System integriert. Mit Hilfe eines Workflows können Mappings definiert, automatisch erzeugt, nachbearbeitet und für die semantische Suche aktiviert werden. Insbesondere der Möglichkeit zur Nachbearbeitung kommt besondere Bedeutung zu. Obwohl die verfügbaren Werkzeuge teilweise sehr gute Trefferquoten vorweisen, sollten automatisch erzeugte Mappings lediglich als initiale Vorschläge aufgefasst, kritisch

hinterfragt und anwendungsspezifisch angepasst werden. Abbildung 16 zeigt ein Beispiel eines Mappings zwischen GEMET (linke Bildhälfte) und der Lebenslagen-Ontologie (rechte Bildhälfte).

Die in der Bildmitte dargestellten automatisch gefundenen Mappings können akzeptiert, verändert, selektiv gelöscht und durch weitere manuell erzeugte Mappings ergänzt werden.

4.3 Energieatlas 2015

Die baden-württembergische Landesregierung verfolgt ehrgeizige Klimaschutzziele. So sollen zum Beispiel der Energiebedarf im Land bis 2050 um die Hälfte reduziert und die dann benötigte Energie zu 80 Prozent aus erneuerbaren Energiequellen gewonnen werden. Nur wenn das gelingt, können die im Klimaschutzgesetz von 2013 (Baden-Württemberg 2013) festgelegten Ziele zur Minderung des Treibhausgasausstoßes um mindestens 25 Prozent bis 2020 und um 90 Prozent bis zum Jahr 2050 erreicht werden.

Eine der in Baden-Württemberg durchgeführten Maßnahmen war die Entwicklung des „Potenzialatlas Erneuerbare Energien“ als landesweite, energiethemenübergreifende Datenbasis zur Bilanzierung von Bestands- und Potenzialdaten. Im Auftrag des Ministeriums für Umwelt, Klima und Energiewirtschaft wurde der Potenzialatlas von der Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg (LUBW) entwickelt und im März 2013 im Rahmen einer Landespressekonferenz für die Öffentlichkeit freigeschaltet.

Anfang 2014 wurde die Weiterentwicklung des Potenzialatlas zum „Energieatlas Baden-Württemberg“ beschlossen. Neben dem Ausbau des bestehenden Informationsangebots zu den Themen Windkraft, Solarenergie und Wasserkraft sollten die neuen Themenbereiche Biomasse und Wärmebedarf sowie Informationen zu Strom- und Gasnetzen, den Bioenergiedörfern und vorbildlichen Energieprojekten im Land in den zu erstellenden Energieatlas integriert werden. Darüber hinaus wurde die Internetanwendung technisch weiterentwickelt und an die Bedürfnisse der zunehmend mobilen Nutzung solcher Angebote angepasst. Der Energieatlas Baden-Württemberg wurde am 13. November 2015 von Umweltminister Franz Untersteller freigeschaltet und ist seitdem unter www.energieatlas-bw.de zu erreichen.

Der Energieatlas Baden-Württemberg wurde auf Basis des Liferay-Portalservers (Liferay 2014) durch ein Konsortium bestehend aus der Landesanstalt für Umwelt, Messungen und Naturschutz (LUBW), der Gesellschaft für Angewandte Hydrologie und Kartographie mbH (AHK), der xdot GmbH und dem Institut für Angewandte Informatik (IAI) konzipiert und umgesetzt. Die verwendeten Softwarekomponenten (Front- und Backend-Komponenten, Einrichtung und Konfiguration der Cloud-Dienste, Darstellungstemplates) stammen dabei im Wesentlichen vom IAI und wurden vom oder unter Anleitung des Autors umgesetzt. Die optische Gestaltung (Liferay-Theme) wurde durch die xdot GmbH umgesetzt. Für die Bereitstellung der Geodaten kam zunächst die Google Maps Engine zum Einsatz, die nach dessen Einstellung durch CartoDB abgelöst wurde. Die inhaltliche Aufbereitung der Daten sowie die Generierung statischer Inhalte wurde durch die Fa. AHK sowie die LUBW durchgeführt.

4.3.1 Ziele und Zielgruppen des Energieatlas

Der Energieatlas richtet sich sowohl an interessierte Bürgerinnen und Bürger, als auch an Fachleute und Entscheidungsträger in Verwaltung, Forschung und Wirtschaft und

stellt wichtige Informationen zum Stand der dezentralen Energieerzeugung und zum regionalen Energiebedarf zur Verfügung. Darüber hinaus bietet er mit seinem landesweiten Überblick Energieberatern, Planern und anderen interessierten Akteuren Hintergrundinformationen und Handreichungen an.

Lokale, kommunale und regionale Planungen können durch den Energieatlas nicht ersetzt werden, insbesondere stellt er keine Planungsgrundlage für die Regional- und Bauleitplanung dar. Vielmehr ist es Ziel des Energieatlas, allen an der Energiewende beteiligten Akteuren Daten und Informationen bereit zu stellen, auf deren Basis Strategien und Maßnahmen zur Erfüllung der gemeinsamen Klimaschutzziele entwickelt werden können.

Der Energieatlas Baden-Württemberg ermöglicht den Nutzern die vorhandenen Informationen mit Hilfe von Karten und Erläuterungstexten auf themenspezifischen Seiten einzusehen. In der Regel werden für jedes Thema zwei Karten angeboten. In der einen Karte werden die bestehenden Energieerzeugungsanlagen standortgenau dargestellt, in der anderen die ermittelten Energiepotenziale abgebildet. Für einzelne Themen dürfen die vorhandenen Daten aus Datenschutzgründen nur in aggregierter Form dargestellt werden, zum Beispiel auf Gemeinde- oder Baublockebene. Neben den Karten werden zu jedem Thema Texte mit Hintergrundinformationen bereitgestellt, in denen zum Beispiel die Bedeutung des Themas für die Energiewende oder der genaue Prozess der Energiegewinnung erläutert werden. Bei Themen, für die eine Potenzialanalyse durchgeführt wurde, werden zusätzlich die Berechnungsmethodik sowie die Art und Qualität der Ergebnisdaten beschrieben.

Mit Hilfe vorkonfigurierter Links kann von jeder Seite des Energieatlas aus das erweiterte Daten- und Kartenangebot aufgerufen werden. Dabei wird der aktuelle Informationskontext beibehalten, indem dort automatisch die zum momentan ausgewählten Thema passenden Inhalte geladen werden und ggf. der gerade betrachtete Kartenausschnitt eingestellt wird.

Das erweiterte Daten- und Kartenangebot des Energieatlas richtet sich in erster Linie an Experten und die Fachöffentlichkeit, hier werden zusätzliche Detailinformationen und spezielle Auswertemöglichkeiten zur Verfügung gestellt. Die Nutzer können sich benötigte Daten individuell zusammenstellen und in Form von interaktiven Karten, Tabellen und Diagrammen visualisieren. Für einige Themen ist es darüber hinaus möglich, die Daten direkt als Excel-Tabellen oder als Shapefiles herunterzuladen.

Abbildung 17: Daten zu bestehenden Windkraftanlagen im erweiterten Daten- und Kartenangebot des Energieatlas (Screenshot)

Abbildung 17 zeigt eine solche Detailansicht mit einer Reihe von Attributen, z.B. zuständige Behörde, Bezeichnung der Anlagenstätte, Anlagenstatus, Hersteller und den Typ der Anlage, für anhand ihrer geografischen Lage ausgewählte Windkraftanlagen. Per Klick lassen sich Kartendarstellungen der angezeigten Anlagen oder Datendownloads aufrufen.

4.3.2 Erscheinungsbild des Energieatlas

Die technische Implementierung des Energieatlas Baden-Württemberg auf Basis der Liferay-Portalplattform (Liferay 2014) erfolgte in enger Zusammenarbeit mit Partnern aus dem INOVUM-Vorhaben (Gschwender et al. 2016). Im Folgenden werden die wichtigsten Konzepte und technischen Hintergründe zur Entwicklung von Layout und Design, Systemarchitektur und den verwendeten Liferay-Portlets erläutert.

Um den genannten Anforderungen wie einfache Bedienbarkeit und mobile Nutzung gerecht werden zu können, wurde das Design des Energieatlas grundlegend überarbeitet. Optisch orientiert es sich an den Webseiten der LUBW und den Vorgaben des Landeslayouts Baden-Württembergs für nachgeordnete Stellen.

Durch die Verwendung derselben Farben und eines ähnlichen Seitenaufbaus mit horizontalem Hauptmenü ist die Nähe des Energieatlas zu LUBW Homepage klar zu erkennen. Vom LUBW Webauftritt aus ist der Energieatlas von den Themen Seiten Erneuerbare Energien direkt verlinkt. Er wirkt somit quasi in die LUBW Seiten integriert, behält aber trotzdem den Charakter eines eigenen Webauftrittes bei.

Das Layout der Seiten ist zweigeteilt und verwendet ein Navigationsmenü auf der linken Seite sowie einen Inhaltsbereich auf der rechten Seite. Die Bereiche sind durch verschiedene Hintergrundfarben klar unterscheidbar. Die Breite des Inhaltsbereiches wurde auf 1000 Pixel erweitert, damit sich Karten möglichst groß darstellen lassen. Der

Navigationsbereich wurde dafür im Vergleich zu den LUBW Webseiten auf 170 Pixel Breite verkleinert.

Ein Merkmal des neuen Layouts ist die Verwendung eines einleitenden Bildes pro Thema über die gesamte verfügbare Bildschirmbreite. Dadurch wird die Seite optisch aufgewertet und zeigt sich moderner. Das Bild wird bei Verwendung von verschieden breiten Endgeräten mit unterschiedlicher Auflösung immer so skaliert, dass die Bildmitte zentriert bleibt und der linke und rechte Rand abgeschnitten werden. Die Bilder wurden im Vorfeld so ausgewählt, dass deren Informationsgehalt dabei nicht verloren geht.

Alle Webseiten sind responsiv programmiert und passen sich der Bildschirmgröße des Endgerätes automatisch an. Beim Zugriff über einen PC wird eine feste Seitenbreite verwendet. Beim Aufruf über ein Tablet oder ein Smartphone werden die Seiteninhalte untereinander in einer Spalte dargestellt. Das Hauptmenü und das Navigationsmenü kollabieren in der mobilen Ansicht und können vom Benutzer bei Bedarf bequem über einen Menübutton aufgeklappt werden.

Abbildung 18: Komponente zur Anzeige von aktuellen Kennzahlen für die Einspeisung von Wind- und Solarenergie (Ausschnitt Screenshot Energieatlas Baden-Württemberg)

Technisch wurden das neue Design und das Layout mit einem neuen Liferay-Theme umgesetzt, das Teile des LUBW-Themes wiederverwendet, aber auch neue Komponenten, z.B. zur Anzeige von Live-Einspeisedaten erneuerbarer Energiequellen (Abbildung 18), enthält. Als CSS-Framework wurde Bootstrap in Verbindung mit AlloyUI und jQuery eingesetzt. Sämtliche Stylesheet Dateien sind in Sassy CSS-Syntax

geschrieben (SCSS) und werden von einem SASS-Preprocessor (Syntactically Awesome Style Sheets) in CSS Dateien konvertiert.

Für die Redakteure wurden außerdem zahlreiche Vorlagen entwickelt (Liferay Webcontent Strukturen und Application Display Templates), mit denen Inhaltselemente identisch darstellt werden können. Alle Projektbeschreibungen für Einzelprojekte und Bioenergiedörfer basieren beispielsweise auf nur einer Vorlage und werden so klar strukturiert und auf sehr übersichtliche Art und Weise angezeigt.

4.3.3 Systemarchitektur

Die technische Implementierung des Energieatlas Baden-Württemberg basiert auf einer serviceorientierten Architektur entsprechend der dritten Architekturvariante (Serviceorientierung, „Webcache“, clientseitige Verarbeitung). Als Frontend-Anwendung kommt die Software Liferay Portal in der Community-Edition zum Einsatz (Liferay 2014). Neben der Pflege redaktioneller Inhalte dient sie vor allem der Integration von Inhalten und Daten aus verschiedenen Quellen. Zahlreiche Komponenten des Energieatlas stammen dabei aus dem LUPO-Portalbaukasten (Schlachter et al. 2014b), insbesondere Komponenten zur Anzeige von Karteninhalten, zur Auswahl von Kartenlayern und zur Anzeige von Objektinformationen. Weitere Synergien ergaben sich aus der gemeinsamen Entwicklung eines Liferay-Themes zur Umsetzung des Corporate Designs des Landes Baden-Württemberg, das in seinem Kern sowohl beim Energieatlas als auch bei der Homepage der LUBW Verwendung findet.

Für die Bereitstellung von Daten kommen verschiedene Hintergrunddienste zum Einsatz, die überwiegend durch eine Cloud-basierte Infrastruktur bereitgestellt werden. Sämtliche Geodaten werden in Form von Kartenlayern durch eine auf CartoDB (CARTO 2017) beruhenden Server-Anwendung bereitgestellt. Originaldaten aus verschiedenen Fachsystemen werden so vom Energieatlas entkoppelt und im „Webcache“ (s. 3.5.1) redundant vorgehalten. CartoDB bietet dabei neben vektor- und kachelbasierten Ansichten der Geodaten auch Programmierschnittstellen (APIs) zum Abruf der Objektdaten, die z.B. für die Breitstellung von Detailansichten genutzt werden.

Andere Daten werden über weitere (Mirco-)Services bereitgestellt, die ebenfalls in der Cloud gehostet werden. Beispiele hierfür sind die aktuellen Kennzahlen für die Einspeisung von Wind- und Solarenergie, die bei den Netzbetreibern abgerufen und anschließend über APIs zur Darstellung im Energieatlas bereitgestellt werden. Die entsprechenden generischen Anzeigekomponenten sind hochkonfigurierbar und Template-basiert. So lassen sich Änderungen in der Regel ohne Programmieraufwand durch Redakteure des Energieatlas durchführen.

Aufgrund der serviceorientierten Architektur werden viele Inhalte des Energieatlas mehrfach verwendet, zum Beispiel stehen viele Kartenlayer ebenfalls im Umweltportal Baden-Württemberg zur Verfügung oder werden in der mobilen App „Meine Umwelt“ zur lokalisierten Information über erneuerbare Energien angeboten. Neue Inhalte und

Dienste lassen sich durch das modulare, auf Web-Widgets bzw. Portlets basierende Konzept leicht in den Energieatlas integrieren.

4.3.4 Weiterentwicklung und Flexibilisierung der Liferay-Portlets

Der bereits im vorigen Abschnitt erwähnte LUPO Portalbaukasten (Schlachter et al. 2011b; Schlachter et al. 2014b), der als Basis für die Entwicklung von Umweltportalen für inzwischen fünf Bundesländer dient, beinhaltet bereits eine ganze Reihe von Komponenten, die im Kern für die Nutzung im Energieatlas Baden-Württemberg geeignet waren. An einigen Stellen bestanden jedoch Anforderungen, die mit den bestehenden Komponenten nicht ohne Ergänzungen umsetzbar waren. Betroffen waren insbesondere die Konfigurationsmöglichkeiten einzelner Anzeigekomponenten, jedoch auch die Möglichkeiten zum Austausch von Informationen zwischen den einzelnen Komponenten.

Zum Informationsaustausch kommunizieren die Frontend-Komponenten des LUPO-Baukastens über einen ereignisbasierten Kommunikationsbus auf Basis eines Publish-Subscribe-Modells (Eugster et al. 2003), d.h. sie melden sich an einer clientseitigen Kommunikationsplattform („Eventbus“) an, und können darüber Nachrichten versenden bzw. empfangen, z.B. bei Nutzerinteraktionen. Klickt ein Nutzer in der Karte auf ein Objekt oder ändert die Zoomstufe und damit den angezeigten Kartenausschnitt, wird ein Ereignis ausgelöst, das an andere UI-Komponenten weitergeleitet wird, die dann autonom darauf reagieren können, z.B. Sachdaten des in der Karte angeklickten Objektes anzeigen.

Die für den Energieatlas gemachten Erweiterungen, z.B. die Anzeige von Detailinformationen beim Klick auf Objekte im Kartenclient oder in einer Liste, flossen an den LUPO-Baukasten zurück und erweiterten so dessen Flexibilität und Funktionsumfang um z.B. die Möglichkeit zur Integration eines Orts-Suchschlitzes in die Kartenkomponente, die Anzeige von Default-Inhalten in der Objektinformation-Komponente, die Bereitstellung des aktuellen Kartenausschnitts als Event für weitere Komponenten oder die Möglichkeit zum orts- und themenscharfen Einsprung in das Fachsystem Umweltdaten und -karten online (UDO). Darüber hinaus wurden die Möglichkeiten zur Attribuierung von Kartenkonfigurationen erweitert, z.B. um Legenden in den Kartenansichten per Vorkonfiguration ein- bzw. auszublenden.

Zusätzlich zur Weiterentwicklung bestehender Komponenten, z.B. des Kartenwidgets, der Kartenlayer-Auswahl und der Objektinfo-Anzeige, wurden einige neue Komponenten spezifisch für den Energieatlas entwickelt, beispielsweise die Anzeige von Kennzahlen für die Einspeisung von Wind- und Solarenergie auf der Startseite (Abbildung 18). Die Komponente ist ein typisches Beispiel für die Umsetzung der serviceorientierten Architektur und damit auch eine Art Blaupause für die Entkopplung der Datenbereitstellung per Service und der Anzeige der Daten im Portal. Die Originaldaten werden dabei in Form von XML-Dokumenten durch den Netzbetreiber bereitgestellt. Ein Update-Service transferiert die Daten zu in der Google Cloud gehosteten relationalen Da-

tenbanken. Ein weiterer Dienst konsumiert die Daten aus der Datenbank und stellt sie in Form einer definierten REST-Programmierschnittstelle als JSON- (JavaScript-) Objekte zur Verfügung, die durch die Anzeige-Komponenten direkt verarbeitet, d.h. Template-basiert dargestellt werden können. Durch die so erreichte Entkopplung wird der Dienst, der die Originaldaten bereitstellt, nur minimal belastet. Die potenziell zahlreichen Anfragen aus dem Energieatlas werden durch eine leistungsfähige, skalierbare Infrastruktur (App Engine und Cloud SQL) in der Cloud bearbeitet. Mögliche Latenzen bei Updates der Originaldaten werden durch eine hinreichend häufige Abfrage der Daten durch den Update-Service minimiert.

4.4 LUPO-Portale

Die derzeit in fünf Bundesländern (Baden-Württemberg, Bayern, Nordrhein-Westfalen, Sachsen-Anhalt und Thüringen) verfügbaren Landesumweltportale (LUPO) stellen einen Baukasten von Komponenten, Diensten und gemeinsamen Technologien dar, aus dem inzwischen auch weitere Projekte wie die mobile App „Meine Umwelt“ (Schlachter et al. 2011a; Schlachter et al. 2013) entstanden sind. Die Mehrfachnutzung von Daten, Diensten und generischen Komponenten in verschiedenen Portalen, Websites, mobilen Apps und weiteren Anwendungen bilden eine grundlegende Säule für die Wirtschaftlichkeit der Software. Im Zuge der Modernisierung des LUPO-Baukastens wurde dabei konsequent auf eine ganze Reihe moderner Technologien sowie eine serviceorientierte Gesamtarchitektur gemäß der dritten Architekturvariante (Serviceorientierung, „Webcache“, clientseitige Verarbeitung) gesetzt, und dabei dennoch ein evolutionäres und agiles Vorgehen (Gründerszene Lexikon 2017) gewählt, um einerseits den Betrieb der Landesumweltportale auch während der Entwicklungsphase sicher zu stellen, andererseits bereits in Zwischenschritten auf neue Technologien, Frameworks und Produkte setzen zu können.

Ein Schwerpunkt des Umbaus war die Einführung von Services, die den Umweltportalen Zugang zu Daten, insbesondere Messwerten, Sachdaten sowie Kartendaten bieten. Dabei kommen auch bewährte Dienste wie die klassische Volltextsuchmaschine zum Einsatz, die nun über eine unabhängige Schnittstelle angebunden ist und durch eine Suchmaschine für strukturierte Daten ergänzt wird.

Im Bereich der Frontend-Komponenten wurde auf eine von konkreten CMS- und Portalsystemen unabhängige Lösung gesetzt. Alle entwickelten UI-Komponenten sind als Web Widgets verfügbar und damit grundsätzlich in beliebigen Webseiten verwendbar (s. 3.5.3). Für die Nutzung in modernen Portalsystemen wie Liferay Portal stehen jedoch sogenannte Wrapper-Portlets für alle Widgets zur Verfügung, welche den Komfort bei der Einbindung und Konfiguration der Widgets innerhalb von Portalen deutlich erhöhen.

Die Klammer für die unabhängig voneinander nutzbaren, generischen Frontend-Komponenten bildet eine Ereignis-basierte Kommunikationsschicht in Form eines Eventbusses. Der Eventbus bietet Kanäle zum Nachrichten- und Datenaustausch unter

den Komponenten und damit auch die Möglichkeit, ein Zusammenwirken von selbständigen Komponenten mit dem Ziel zu erreichen, dem Nutzer eine schlüssige Gesamtanwendung präsentieren zu können. Um die Anforderungen an die Daten und Dienste dabei möglichst gering zu halten, z.B. um bestehende Dienste und Datenbestände einbeziehen zu können, wurde zunächst auf eine relativ lose Kopplung auf Basis von allgemeinen Nachrichten (Ortsbezug, Themenbezug, Suchbegriffe) gesetzt.

Auch bei der Umsetzung der konkreten Landesumweltportale auf Basis der neuen Technologien wurde ein schrittweises Vorgehen gewählt. Während das neue Landesumweltportal Baden-Württemberg bereits Ende November 2014 online ging, wurden für die anderen Länder zunächst Demonstratoren („Showcases“) entwickelt, die anschließend sukzessive in produktive Portale überführt wurden bzw. werden. Dabei kam es zum Umbau wesentlicher Komponenten, z.B. die Ablösung der Google Maps Engine als Hosting-Lösung für Karten und Geoinhalte durch die ebenfalls in der Cloud betriebene Software CartoDB (CARTO 2017).

Die Landesumweltportale wurden seit dem Jahr 2003 als Kooperationsprojekt des Ministeriums für Umwelt, Klima und Energiewirtschaft Baden-Württemberg (UM), der Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg (LUBW) und dem Institut für Angewandte Informatik (IAI) als technischem Entwicklungspartner konzipiert und implementiert. Im Lauf der Zeit entstand dabei eine länderübergreifende Kooperation der oben aufgeführten Länder. Die Architektur und Umsetzung der Landesumweltportale sowie der ihnen zugrundeliegenden Dienste wurde dabei federführend durch den Autor der vorliegenden Arbeit erarbeitet und weiterentwickelt, die Implementierung aller Komponenten erfolgte bis zum Jahr 2013 im Wesentlichen ebenfalls durch den Autor. Mit dem Umstieg auf die Basissoftware Liferay Portal kam die Firma xdot GmbH als weitere Entwicklungspartner hinzu. xdot beteiligt sich seitdem hauptsächlich bei der Entwicklung im Bereich Design (Liferay Themes) und beim Betrieb der Portale (Hosting). Der Autor verantwortet und koordiniert seitdem die Umsetzung der Frontend- und Backendkomponenten, die teilweise auch von Kollegen des IAI sowie teilweise im Rahmen studentischer Arbeiten implementiert wurden.

Diskussion der Event-basierten Kommunikation von Komponenten

Für die Landesumweltportale hat die Event-basierte Kommunikation mit einer losen Kopplung von (Umwelt-)Objekten einen entscheidenden Vorteil: Sie reduziert den Aufwand bei der Einbindung von Umweltdaten in die Landesumweltportale auf ein leistbares Niveau. Die Landesumweltportale bieten auf der einen Seite Zugang zu einer äußerst heterogenen Landschaft von Umweltdaten:

- Struktur (strukturierte, semistrukturierte, unstrukturierte Daten)
- Ortsbezug (Daten mit und ohne expliziten Ortsbezug in einer Vielzahl von Repräsentationen)

- Daten aus unterschiedlichen technischen Systemen mit einer Vielzahl von Schnittstellen und technischen Formaten, verschiedenen IDs oder Schlüssellisten etc.

Die Daten- bzw. Systemlandschaft bietet in den meisten Fällen keine expliziten Beziehungen zwischen Daten und Objekten bzw. zumindest keine technisch nutzbare Umsetzung dafür.

Auf der anderen Seite erwarten die menschlichen Nutzer der Umweltportale in den meisten Fällen zwar eine Unterstützung beim Auffinden der passenden Informationen zu ihrem Anliegen, stellen dabei aber selbst eine aktive Filterinstanz dar, welche die angezeigten Informationen sichten, bewerten und sich passende Teile herauspicken kann. Eine – nicht zu große – Obermenge der tatsächlich relevanten Ergebnisse ist für sie in den meisten Fällen akzeptabel. Des Weiteren hat sich gezeigt, dass bei einem großen Anteil der Suchanfragen die Beziehungen zwischen den passenden Ergebnissen auf einem sehr hohen Abstraktionsniveau darstellbar sind, z.B. ihrer örtliche Nähe zueinander. So kann beispielsweise die Frage, ob die Windkraftanlagen einer Gemeinde innerhalb oder in der Nähe von Naturschutzgebieten liegen, mit Hilfe der Umweltportale sehr leicht beantwortet werden: Allein durch Eingabe der Suchbegriffe „windrad schutzgebiet langenburg“ erhält der Nutzer bereits die gewünschten Informationen, allerdings tatsächlich mehr als verlangt, da neben Naturschutzgebieten auch Objekte anderer Schutzgebietstypen (Biotope, Nationalparke etc.) dargestellt werden.

Durch das einfache Abwählen der nicht benötigten Schutzgebietstypen lässt sich die obige Frage klären, denn alle Naturschutzgebiete und Windkraftanlagen im Bereich der Gemeinde Langenburg werden angezeigt. Zwar muss der Nutzer die Beziehung zwischen Windkraftanlagen und Schutzgebieten noch selbst herstellen, allerdings gelingt das dank Kartenansicht „auf einen Blick“ – obwohl in der verwendeten Datengrundlage eine Beziehung wie „liegt in/bei“ nicht vorhanden ist; im Gegenteil: Informationen über Windkraftanlagen und Naturschutzgebiete kommen aus völlig unterschiedlichen Systemen und sind nur über die gemeinsame Darstellung innerhalb des Kartenclients miteinander verbunden.

Die lose Kopplung von Umweltobjekten funktioniert also nur für menschliche Nutzer und nur in solchen Anwendungsfällen, in denen sich Beziehungen auf relativ hohen Abstraktionsniveaus darstellen lassen, z.B. ihre örtliche Nähe oder die Zuordenbarkeit zu einem bekannten Thema (Windrad ➔ Windkraft).

4.5 Mobile Apps

„Meine Umwelt“ und „Meine Pegel“ sind Apps für mobile Endgeräte und bieten Zugriff auf behördliche Umweltinformationen bzw. Pegelstände. Dabei nutzen beide Apps spezifische Möglichkeiten der Mobilgeräte aus, z.B. die automatisierte Bestimmung des Standorts per GPS-Sensor, die Möglichkeit zur aktiven Aufzeichnung und Meldung von Umweltinformationen per Kamera oder Mikrofon oder die Möglichkeit zur proaktiven Benachrichtigung des Nutzers per Push-Notification.

Sie werden im Rahmen einer Entwicklungskooperation, bestehend aus der Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg (LUBW), der Firma xdot GmbH sowie dem Institut für Angewandte Informatik (IAI) des Karlsruher Instituts für Technologie (KIT) unter Federführung des Ministeriums für Umwelt, Klima und Energiewirtschaft Baden-Württemberg im Rahmen der länderübergreifenden Entwicklungskooperation Landesumweltportale (LUPO), entwickelt, betrieben und betreut (Schlachter et al. 2012; Schlachter et al. 2013).

Die grundlegende Architektur der App „Meine Umwelt“ als hybride App unter Nutzung von Diensten des Webcache sowie die ersten beiden prototypischen Implementierungen der App stammen dabei vom Autor der vorliegenden Arbeit. In das Konzept der App sind auch Bestandteile der zweiten Architekturvariante (Zielsystembeschreibungen zur Konfiguration generischer Bausteine) eingeflossen. Nachdem die Untersuchungen zur Grundarchitektur, zu möglichen Technologien und zur Funktionalität der App abgeschlossen waren, wurden die weiteren Implementierungsarbeiten sowie die Umsetzung des Designs von der xdot GmbH übernommen, die die App bis heute weiterentwickelt. Der Autor ist seitdem als Systemarchitekt und als Leiter des Umsetzungsteams des IAI an der Weiterentwicklung der App und der benötigten Datendienste beteiligt.

Im Folgenden wird zunächst die App „Meine Umwelt“ vorgestellt und dann auf die Funktionen der neuen App „Meine Pegel“ eingegangen. Schließlich werden die gemeinsamen Grundlagen beider Apps im Rahmen des App-Baukastens „LUPO mobil“ (für „Landesumweltportale mobil“) dargestellt.

4.5.1 App „Meine Umwelt“

Die zentrale Idee der App „Meine Umwelt“ ist es, in einer einzelnen App verschiedene Umwelt-bezogene Anwendungsfälle zusammenzufassen. Abbildung 19 zeigt drei der Anwendungsfälle. Dazu gehören die Bereitstellung von Umweltinformationen („Informieren“, links), das Sammeln neuer bzw. das Aktualisieren vorhandener Umweltinformationen („Melden“, mittig) sowie das Bereitstellen lokalisierter Informationen für die Orientierung und Nutzung vor Ort („Erleben“, rechts).

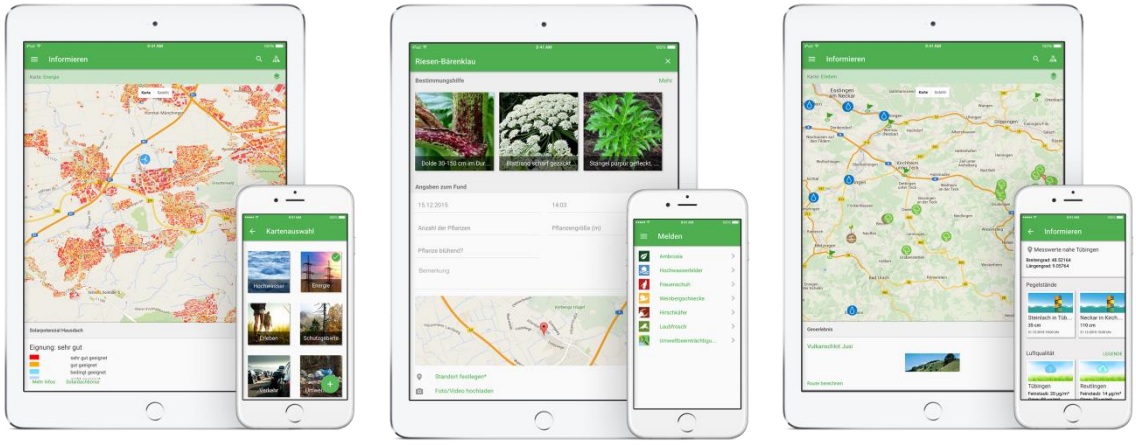

Abbildung 19: Informieren (links), Melden (mittig), Erleben (rechts) – Kernfunktionen der „Meine Umwelt“-App (Screenshots der App „Meine Umwelt“)

Mit Hilfe der App ist es möglich, sich standortgenau über Messwerte zur Luftqualität, zu aktuellen Wasserständen sowie über Umweltdaten aus unterschiedlichen Themenbereichen wie Schutzgebiete, Verkehr, Energie oder Hochwassergefahrenkarten zu informieren. Darüber hinaus können von den Nutzern der App gemeldete Artenfunde und Umweltbeeinträchtigungen abgerufen werden. Zusätzlich findet man Informationen zu Naturdenkmälern und Erlebnisorten. Zurzeit kann die App in Baden-Württemberg, Sachsen-Anhalt und Thüringen verwendet werden. Der Daten- und Funktionsumfang ist vom gewählten Bundesland abhängig und kann daher regional unterschiedlich sein. Die Ausweitung auf weitere Bundesländer über die Integration von bundesweiten Themen sowie über die Aufnahme weiterer Partner in der LUPO-Kooperation ist in Planung.

In Abbildung 20 werden der Startbildschirm, das Navigationsmenü sowie das Menü zur Auswahl des Bundeslandes dargestellt. Wird ein anderes Bundesland ausgewählt, so passen sich die verfügbaren Kartenthemen, Messwerte und Meldethemen entsprechend an. Ein automatisches Setzen des Bundeslandes auf Basis des Standortes wurde mehrfach diskutiert. Die Genauigkeit der GPS-Informationen an den Grenzen eines Bundeslandes und die Möglichkeit explizit in die Themen eines anderes Bundesland springen zu können, sprechen aber dafür, die manuelle Einstellung im Sinne der Personalisierbarkeit auf ein bestimmtes Bundesland zu belassen. Die Möglichkeit, die App spezifisch zu regionalen Inhalten auf ein Bundesland anzupassen wird aber weiterhin durch das Mitwirken an der LUPO mobil-Kooperation möglich sein.

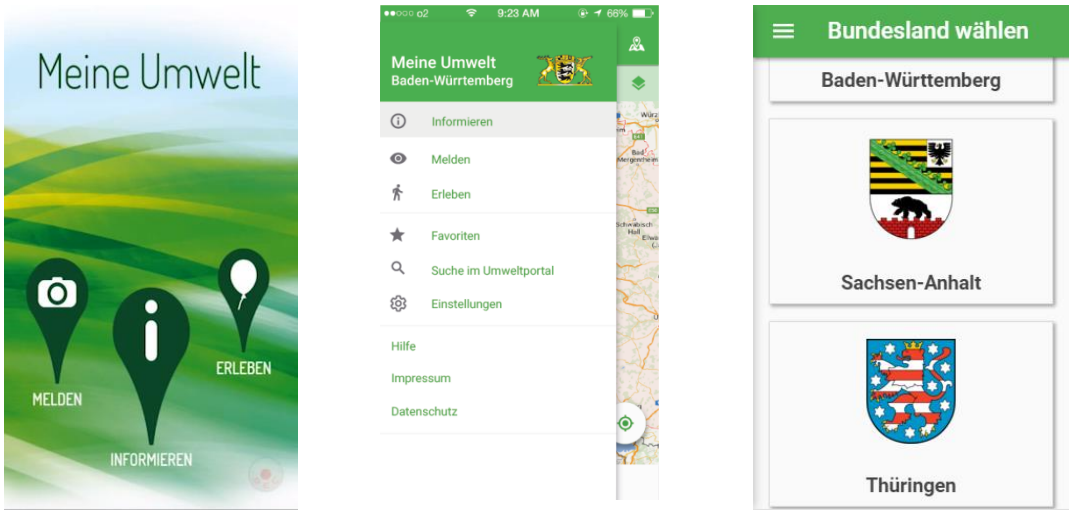

Abbildung 20: Start-Bildschirm, Navigation und Auswahl des Bundeslandes (Screenshots der App „Meine Umwelt“)

Im Bereich „Informieren“ kann man bestimmte Karteninhalte thematisch auswählen (Abbildung 21). Wählt man die Karte „Energie“, so kann man sich standortgenau bspw. über Unterthemen wie Solarpotentialflächen, Windkraftanlagen und Energieagenturen informieren. Wählt man das Thema „Schutzgebiete“, so stehen die Layer Naturdenkmäler, Naturschutzgebiete, Wasserschutzgebiete, Natura 2000-Flächen und Landschaftsschutzgebiete zur Verfügung. In der Kartenansicht sind alle Themen initial als Kartenschichten (Layer) sichtbar. Einzelne Layer lassen sich aber zur Verbesserung der Übersicht auch individuell an- und ausschalten.

Neben den Kartendiensten enthält der Bereich „Informieren“ auch Darstellungen aktueller Messwerte (Abbildung 21 rechts). Dem Benutzer werden hier Pegelstände für umliegende Gewässer bzw. Luftqualitätsdaten von Messstationen in seiner Nähe präsentiert. Zusätzlich besteht in den Ländern Thüringen und Sachsen-Anhalt die Möglichkeit zur standortbezogenen Anzeige von klimatischen Kennzahlen. Basierend auf einem 1x1 km Raster repräsentieren sie die wichtigsten statistischen Klimadaten der letzten dreißig Jahre, z.B. die Höchst-, Mittel- und Tiefsttemperaturen. Daneben existieren Informationen zu der Anzahl heißer Tage, Frosttage, Sommertage und Eistage. Durch den Bereich „Informieren“ werden damit gebündelt verschiedene umweltbezogene Informationsbedürfnisse des Anwenders adressiert.

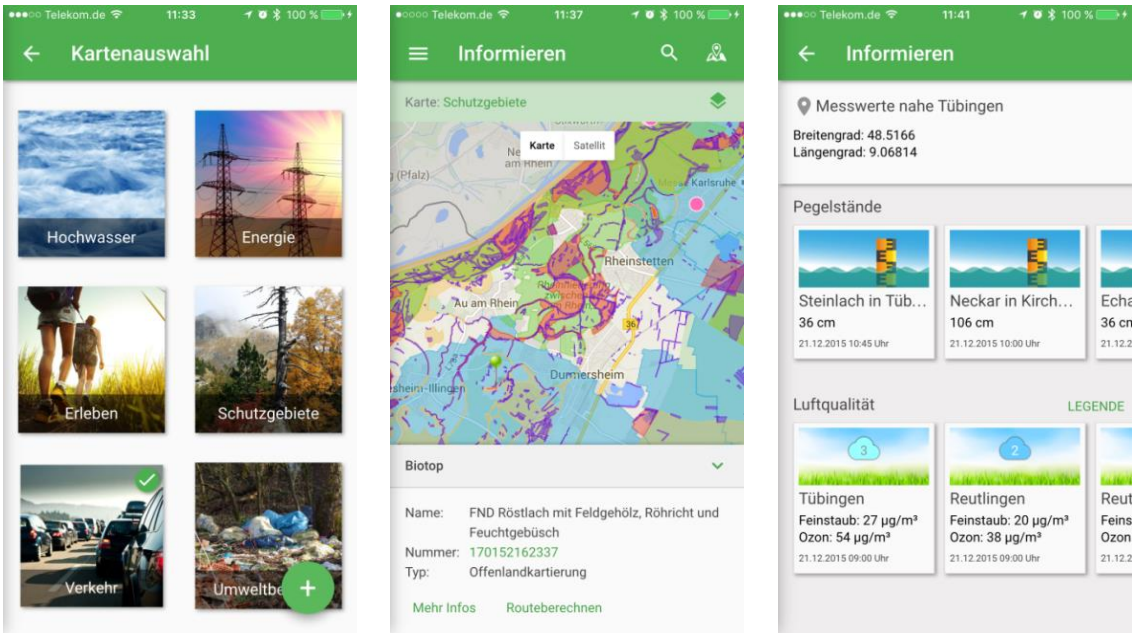

Abbildung 21: Bereich Informieren beinhaltet Karten mit Unterthemen (links), Detailinformationen zu ausgewählten Objekten (mittig) sowie aktuelle Messwerte (rechts) (Screenshots der App „Meine Umwelt“)

Als zweiter Eintrag in der Navigation der App ist der Bereich „Melden“ zu finden (Abbildung 20 mittig). Hiermit wird es den Bürgern ermöglicht, aktiv den Bestand an Umweltdaten zu vergrößern und daran mitzuwirken, deren Qualität und Abdeckungsgrad zu erhöhen. Aufgrund der dahinter liegenden organisatorischen Prozesse sind die Meldethemen pro Bundesland unterschiedlich. Derzeit können in Baden-Württemberg Hochwasserbilder aufgenommen, Funde der seltenen Arten Laubfrosch, Weinbergschnecke, Hirschkäfer und Frauenschuh, die Art Feuersalamander als Lurch des Jahres 2016 sowie Ambrosia-Standorte und Umweltbeeinträchtigungen gemeldet werden (Abbildung 22 links).

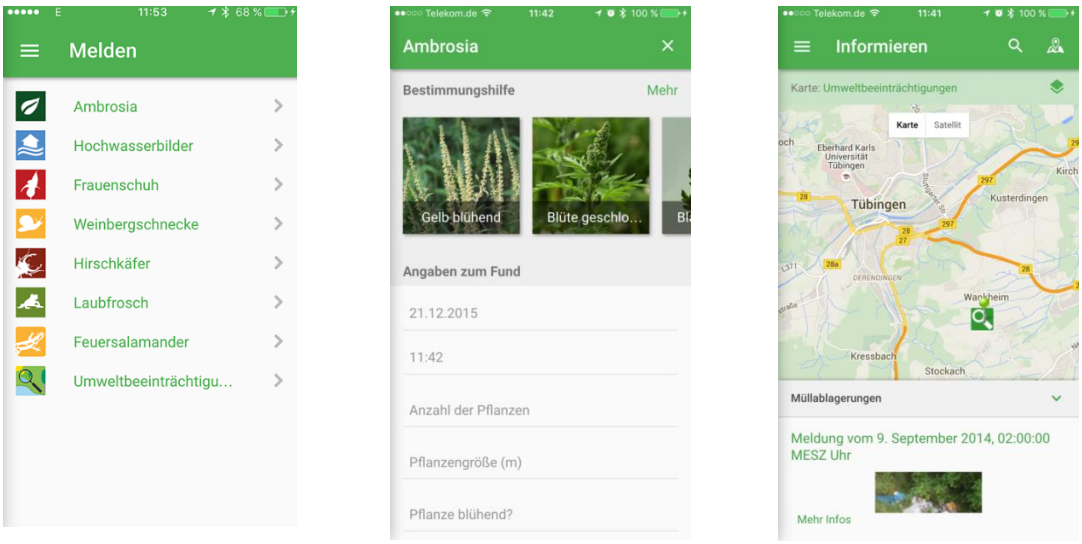

Abbildung 22: Verschiedene Meldethemen (links), Formular zum Erfassen von Standort, Sachdaten (mittig) im Bereich „Melden“, sowie die Anzeige von eingegangenen Meldungen im Bereich „Informieren“ (Screenshots der App „Meine Umwelt“)

In Thüringen können Hirschkäferfunde und in Sachsen-Anhalt Ambrosia- und Riesenbärenklau-Standorte gemeldet werden, was die Umweltverwaltung in der Kartierung invasiver Arten unterstützt. Zur Identifikation von Arten durch den Nutzer und zur Vermeidung von Falschmeldungen stehen Bestimmungshilfen zur Verfügung. Weitere Meldeprojekte sind in Vorbereitung.

4.5.2 LHP-App „Meine Pegel“

Die App „Meine Pegel“ ist ein Service des länderübergreifenden Hochwasserportals (LHP), in dem die Bundesländer länderspezifische Hochwasserinformationen und Messwerte bündeln und in einer Gesamtübersicht darstellen[20]. „Meine Pegel“ ist die amtliche Wasserstands- und Hochwasser-Informations-App mit Zugang zu den Messwerten von mehr als 1.600 Pegeln in Deutschland. Die App ist für Android, iOS und Windows Phone erhältlich und ermöglicht einen schnellen Überblick zu aktuellen Wasserständen an Pegeln sowie eine kostenfreie Benachrichtigung bei Über-/ oder Unterschreitung von individuell konfigurierbaren Pegelständen. Damit ermöglicht die App es dem Bürger, sich sowohl einen schnellen Überblick zur überregionalen Hochwasserlage in Deutschland und zu den Hochwasserinformationen der Bundesländer einzuholen als auch sich gezielt individuell benachrichtigen zu lassen.

Die App ist somit eine länderübergreifende Anwendung für Informationen, die von verschiedenen Institutionen erhoben und bereitgestellt werden. Von den Hochwasserzentralen der Bundesländer kann für jeden Pegel im jeweiligen Zuständigkeitsbereich per XML einzeln und zeitnah konfiguriert werden, ob und ggf. welcher Informationsumfang hierzu in der App freigeschaltet wird (dezentrale Konfiguration des Informationsumfangs der App). Der Betrieb und die Fortschreibung der LHP-App wird aus Mitteln des Länderfinanzierungsprogrammes „Wasser, Boden und Abfall“ gefördert.

In Abbildung 23 werden die Startseite des Portals zum Stand des Hochwassers im Juni 2013 sowie das für Nutzung auf mobilen Endgeräten optimierte Layout des Portals gezeigt.

[20] https://www.hochwasserzentralen.de

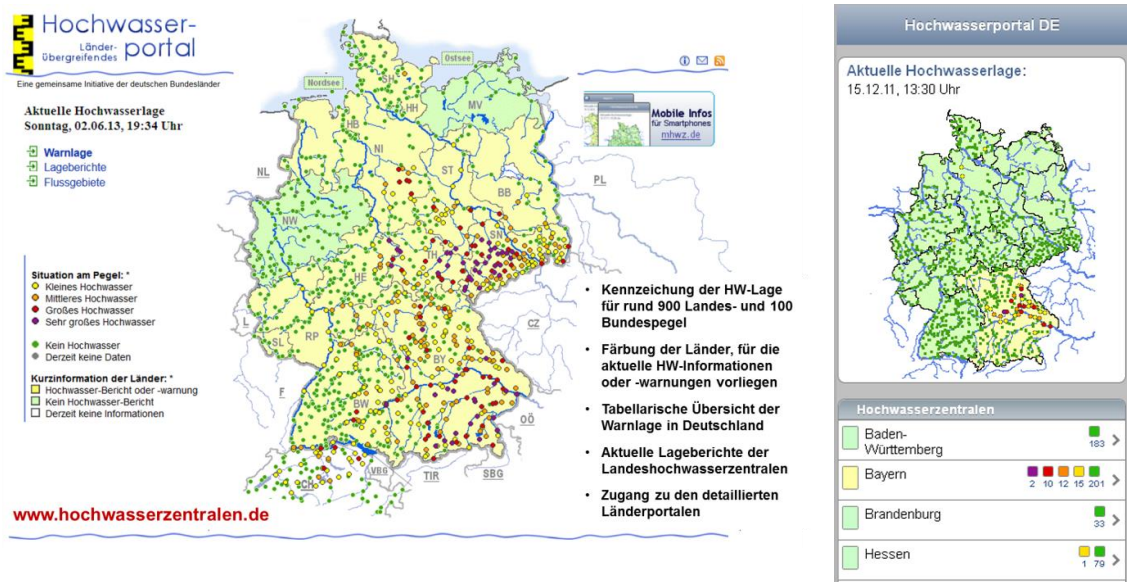

Abbildung 23: Webangebot des länderübergreifenden Hochwasserportals LHP, links der normalen Webansicht, rechts der mobilen Ansicht (Screenshots)

Wie man an der Abbildung sieht, ist es bereits mit Hilfe der für Mobilgeräte optimierten Portalansicht möglich gewesen, sich über aktuelle Wasserstände und die Hochwasserlage allgemein zu informieren. Die App „Meine Pegel“ bietet darüber hinaus die Möglichkeit zur Personalisierung, d.h. die Funktion zum Zusammenstellen von individuellen Favoritenlisten und das Einstellen individueller Schwellwerte zum Auslösen von aktiven Benachrichtigungen auf das mobile Endgerät bei Über- oder Unterschreitung des entsprechenden Wertes am Pegel. Ein Anwender muss sich die Information zum Hochwasser von daher nicht mehr selbst einholen, sondern erhält sie im „Push-Verfahren“ (Urban Airship 2016).

Abbildung 24 zeigt einige wesentliche Screenshots der App „Meine Pegel“. Links ist die Pegelkarte für ganz Deutschland zu sehen, in der die einzelnen Bundesländer je nach Hochwasserlage entsprechend eingefärbt sind und der Status einzelner Pegel angezeigt wird. Über den darunter liegenden Navigationsbereich gelangt man in die Pegelverzeichnisse der einzelnen Bundesländer, die man nach eingetretener Hochwasserklasse filtern bzw. nach dem Namen eines Pegels oder eines Gewässers durchsuchen kann. Hat man einen Pegel ausgewählt, so gelangt man auf die Pegeldetail-Ansicht, in der aktuelle Messwerte wie der Wasserstand sowie der Abfluss dargestellt werden. Darunter befindet sich eine Grafik, die den Verlauf des Wasserstandes der letzten Tage visualisiert. Je nach Bereitstellung durch die zuständige Hochwasserzentrale befindet sich darunter eine Grafik, die Vorhersagen zur weiteren Wasserstandsentwicklung darstellt

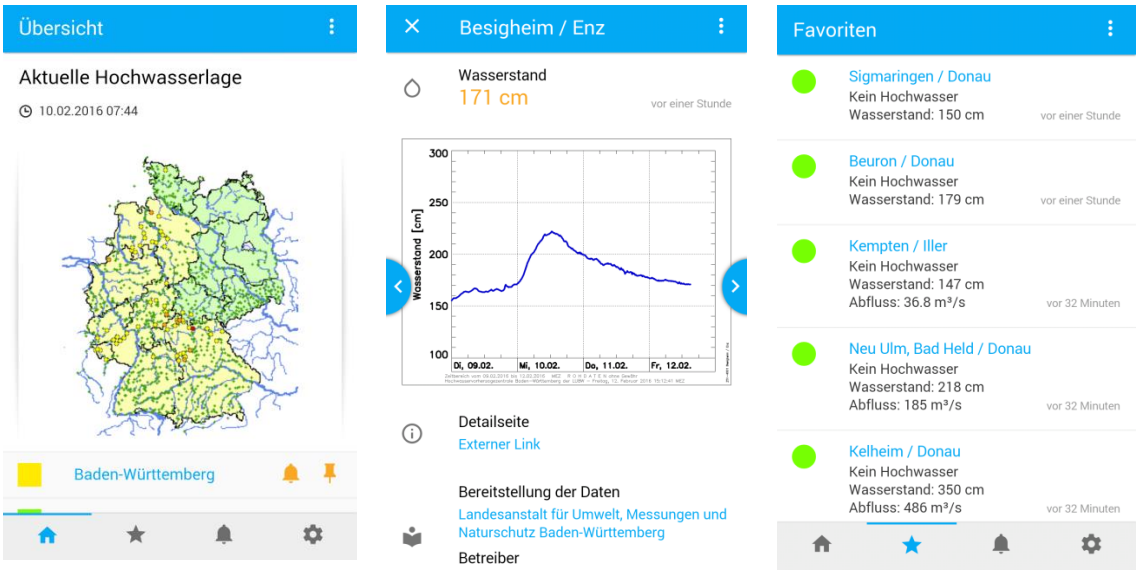

Abbildung 24: Übersicht der Pegel als Karte (links), Pegeldetails mit Ganglinie (mittig) und Favoritenliste (rechts) (Screenshots der App „Meine Pegel“)

In der Detailansicht ist es möglich, den gewählten Pegel in die eigene Favoritenliste zu übernehmen, die in der Abbildung rechts dargestellt ist. Weitere Funktionen, die in der Detailansicht verfügbar sind, werden in Abbildung 25 dargestellt, z.B. die Einrichtung einer Warnung bei einem bestimmten Pegelstand (links), die Einrichtung von Nachrichtenabonnements (Mitteilungen) zu bestimmten Pegeln (mittig) sowie der Empfang solcher Nachrichten (rechts).

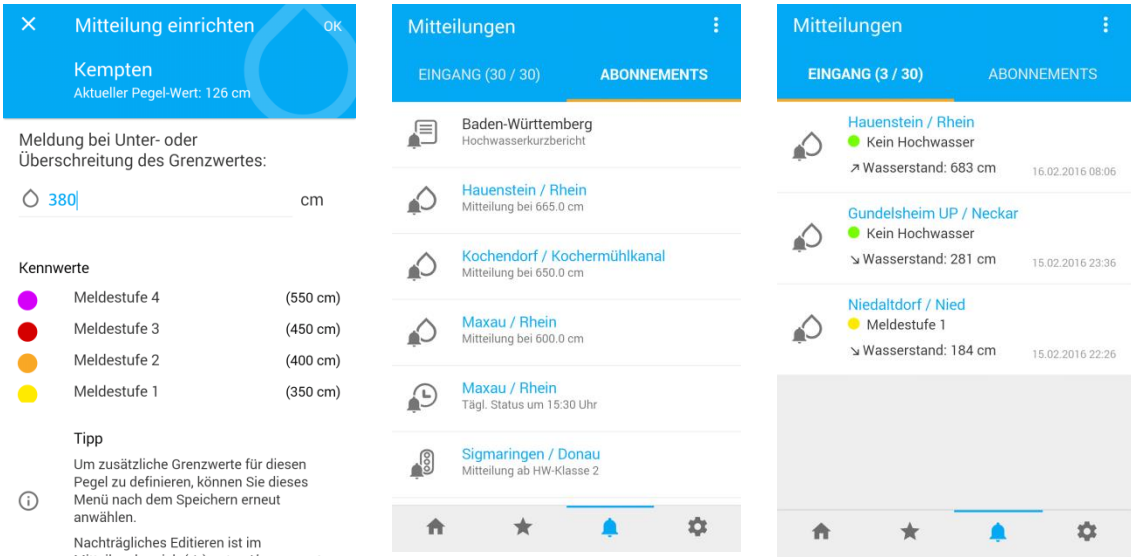

Abbildung 25: Einrichtung einer Pegelwarnung (links), Einrichtung von Abonnement (mittig) und Eingang von Mitteilungen (rechts) (Screenshots der App „Meine Pegel“)

Hier ist es möglich, einen individuellen Grenzwert für den Pegel einzutragen, bei dessen Über- oder Unterschreitung eine Benachrichtigung auf das vorliegende mobile Endgerät erfolgt. Über die Hauptnavigation im unteren Bereich gelangt man zur Ansicht

„Mitteilungen“, die bis zu dreißig Benachrichtigungen für den Anwender speichert und das Verwalten der Abonnements realisiert. Hierdurch können Grenzwerte oder Hochwasserklassen, ab denen eine Benachrichtigung erfolgen soll, angepasst werden. Eintreffende Benachrichtigungen werden initial durch die native Darstellung des jeweiligen Betriebssystems angezeigt und sind damit in ihrem Aussehen vergleichbar mit den gängigen Instant-Messaging Systemen. Durch die Navigation aus der nativen Push-Nachrichtenansicht heraus gelangt man in die App „Meine Pegel“ selbst.

Die App lässt sich auch mit sogenannten „wearable devices“ (Heise online 2017) wie einer Smartwatch koppeln. Unterstützt werden Android Wear und die Apple Watch. Hierbei erhält der Anwender die Benachrichtigung direkt auf die Uhr an seinem Handgelenk und kann bei Bedarf detaillierte Informationen auf dem Smartphone anschauen.

Die App ist seit März 2016 stufenweise in den verschiedenen App Stores veröffentlicht worden und hat seither sehr positive Resonanz erfahren. Wichtige Punkte zur Weiterentwicklung sind der kontinuierliche Ausbau des Datenangebots, die Erhöhung der Redundanz und Ausfallsicherheit im Backend speziell im Falle eines Hochwassers, die Optimierung hin zu einer bundesweiten Suchfunktion für Pegel sowie das Einstellen eines individuell definierbaren Warntons. Anhand eines spezifischen Warntons soll der Anwender die Möglichkeit haben, die Mitteilung von anderen, ggf. weniger wichtigen Warnhinweisen, auditiv unterscheiden zu können.

4.5.3 Technischer Rahmen zur App-Entwicklung

Die in den vorherigen Abschnitten beschriebenen Apps „Meine Umwelt“ und „Meine Pegel“ haben zwar einen unterschiedlichen Funktionsumfang, eine andere Zielgruppe und benötigen unterschiedliche Backend-Dienste als Datenlieferant, dennoch ist es auf Basis eines Rahmenwerks unter Nutzung einer Microservice-basierten Architektur und moderner Webfrontendtechnologien gelungen, die Apps in ihrem strukturellen Aufbau, in ihrer Ereignisverarbeitung sowie in ihren Zugriffsmustern auf die dahinterliegenden Server nach einheitlichen Entwurfsmustern generisch zu realisieren, was schematisch in Abbildung 26 veranschaulicht wird.

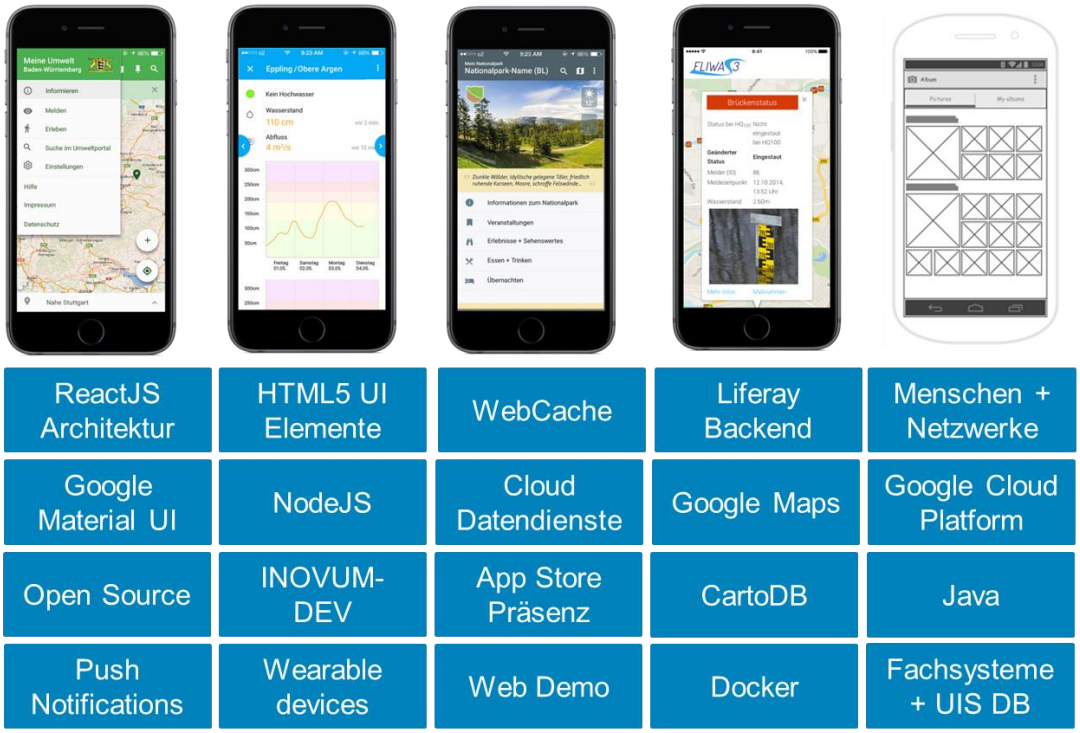

Abbildung 26: LUPO-Baukasten als Fundament zur Erstellung von Umwelt-Apps

Durch den LUPO Baukasten konnte die Entwicklung deutlich wirtschaftlicher gestaltet werden, da eine Wiederverwendung von Vorgehensweisen und Quellcode erreicht werden konnte. Beide Apps setzten in ihrer UI-Konzeption auf die Material UI Design Guidelines (material.io 2017), die Google sehr intensiv und detailliert erarbeitet hat und sich sowohl auf Webanwendungen als auch bei nativen Apps einsetzen lassen. Material Design hat sich an iOS orientiert, bildet das Fundament für die Entwicklung moderner Android-Apps und setzt sich mittlerweile auch für Webanwendungen im Desktop-Bereich nach und nach durch. Auf dem Fundament des „LUPO mobil“-Baukastens zur Erstellung von Umwelt-Apps aufbauend, können auch weitere Apps wirtschaftlich realisiert werden.

Die Fragmentierung an Plattformen, Geräten, Programmiermodellen und Diensten im Bereich Mobile ist sehr stark und unterliegt kontinuierlichen Veränderungen. Den aus der Anzahl verschiedener Plattformen (z.B. Android, iOS, Windows Phone) resultierenden Mehraufwand zur Entwicklung und Betrieb der Apps kann man dabei durch Cross-Plattform Entwicklung von sogenannten Hybrid-Apps (Schlachter et al. 2012) reduzieren. Bei Hybrid-Apps handelt es sich um WebApp‘s auf Basis von HTML5 und JavaScript, die über einen Container, wie ihn z.B. Cordova (Apache Cordova 2017) liefert, als native App bereitgestellt werden. Bei der App Meine Umwelt handelt es sich um eine solche Hybrid-App. Für den HTML-basierten Teil kommen aktuelle Web-Frameworks wie z.B. React (React 2017) zum Einsatz.

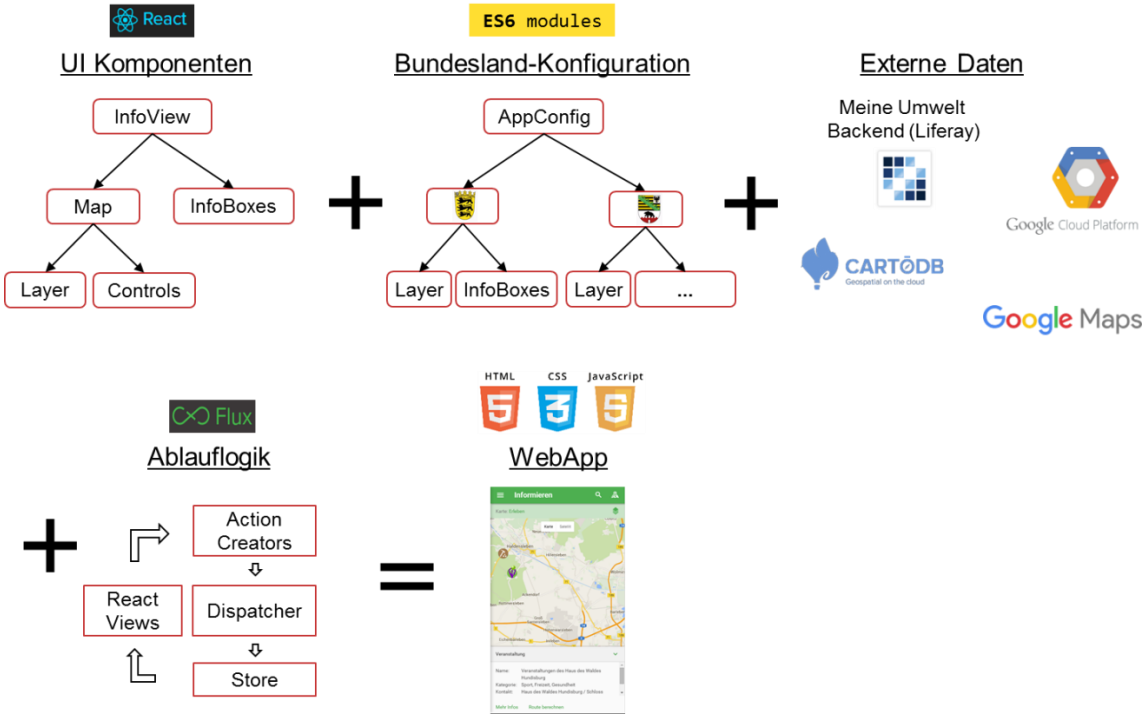

Abbildung 27: Konzeptionelle Struktur der "Meine Umwelt" App

Bei der Konzeption von der Meine Umwelt App waren Wiederverwendbarkeit und Erweiterbarkeit wichtige Ziele. Daher wurde z.B. in der Meine Umwelt App auf das Framework react.js (React 2017) gesetzt, welches es erlaubt, die graphischen Elemente als Komponenten zu realisieren. Hierdurch wird im speziellen die Wiederverwendbarkeit adressiert, da sich die Komponenten in weiteren Apps entweder direkt integrieren oder als Vorlagen nutzen lassen. Die Konzeption ist in Abbildung 27 dargestellt.

Neben der Wiederverwendbarkeit ist bei der Meine Umwelt App vor allem wichtig, dass sie sich einfach mit neuen Daten sowie neuen Bundesländern erweitert lässt. In der App sind daher die verschiedenen Anwendungsfälle Information, Melden und Erleben generisch implementiert. Die eigentlichen Inhalte wie z.B. Karten Layer, Info-Boxen oder Meldearten sind in eigene, bundeslandabhängige Konfigurationsmodule ausgelagert. Hierdurch können neue Daten sowie auch neue Bundesländer verhältnismäßig einfach integriert werden.

Da die App auf Web-Technologien basiert, ist es möglich, auch eine Variante für den Browser über das Internet bereitzustellen (WebApp). Die WebApp wird hauptsächlich zu Test- und Demozwecken verwendet. Sie unterliegt allerdings der Einschränkung, dass die nativen Funktionen der mobilen Endgeräte, wie z.B. der Zugriff auf die Kamera, nicht zur Verfügung stehen. Solche Funktionen können erst in der nativen App über das Framework Cordova verwendet werden.

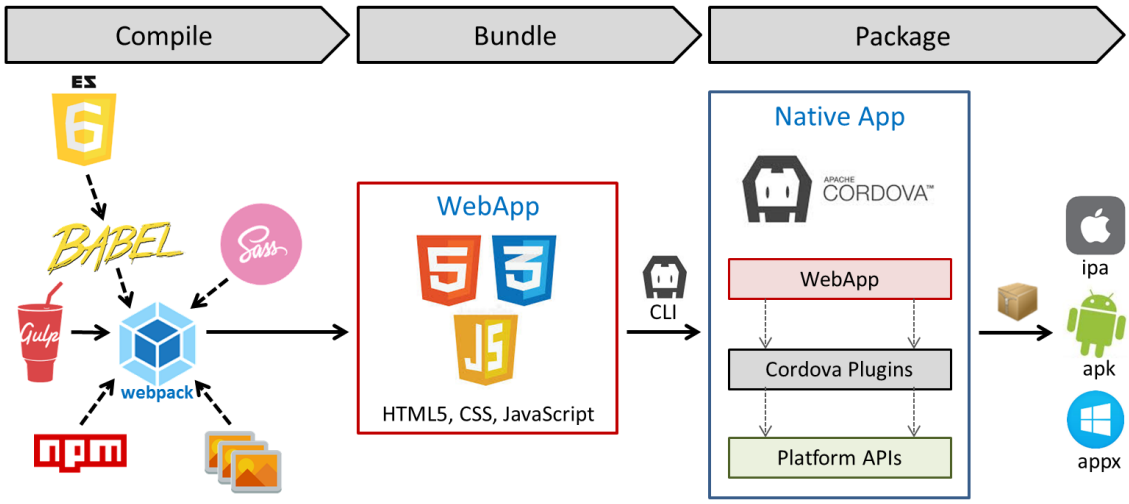

Abbildung 28: Buildpipeline der App „Meine Umwelt“ (nach Projektdokumentation „Meine Umwelt“, xdot GmbH)

Um eine möglichst agile Entwicklung (Gründerszene Lexikon 2017) zu ermöglichen, ist der Buildprozess zu einem hohen Grad automatisiert (Abbildung 28). Zusätzlich sollen bei der Entwicklung neueste Sprachfeatures von JavaScript (ECMAScript 6, ES 6) eingesetzt werden können. Es wurde eine Buildpipeline entwickelt, in der verschiedene Build-Werkzeuge wie Gulp.js, Webpack und Bable.js zum Einsatz kommen (Abbildung 28 links). Konzeptionell ist die Buildpipeline in drei Schritte aufgeteilt. Im „Compile“-Schritt (links) wird der Quellcode aus JavaScript, SASS etc. kompiliert. Im „Bundle“-Schritt (mittig) werden die einzelnen Bestandteile zu einer vollständigen, im Browser lauffähigen, WebApp zusammen geführt. Im Schritt „Package“ (rechts) wird die WebApp dann schlussendlich über Cordova als native App für die verschiedenen Plattformen verpackt. Das Ergebnis sind binäre Pakete, die in die jeweiligen App Stores hochgeladen werden können.

Da die WebApp auf Webtechnologien basiert, ist es möglich, sie für die Nutzung per Webbrowser über das Internet bereitzustellen, was hauptsächlich zu Test- und Demozwecken dient. Die WebApp unterliegt allerdings Beschränkungen, da die nativen Funktionen der mobilen Endgeräte, wie z.B. die Kamera, im Browser nicht zur Verfügung stehen. Solche Features können erst in der nativen App über das Framework Cordova (Apache Cordova 2017) genutzt werden.

Mit der App „Meine Umwelt“ können Bürger in Thüringen, Baden-Württemberg und Sachsen-Anhalt ihre Umwelt besser kennen lernen und aktuelle Umweltdaten mobil abrufen. Die App richtet sich an alle, die spontan vor Ort mehr über ihre Umgebung erfahren möchten. Beispiele sind das Entdecken von Attraktionen in der Umgebung von zu Hause oder unterwegs sowie das Informieren über die Luftqualität, Pegelstände, Umweltzonen sowie das Hochwasserrisiko bzw. das Solarpotential des Wohnortes des Nutzers. Die App „Meine Pegel“ ist die amtliche Wasserstands- und Hochwasser-Informations-App mit mehr als 1.600 Pegeln in Deutschland. Sie ermöglicht einen

schnellen Überblick zu aktuellen Wasserständen an Pegeln sowie kostenfreie Benachrichtigung bei Über-/ oder Unterschreitung von individuell konfigurierbaren Pegelständen. Der Bürger kann sich einen schnellen Überblick zur überregionalen Hochwasserlage in Deutschland und zu den Hochwasserinformationen der Bundesländer einholen. Beide Apps basieren auf dem „LUPO mobil“-Baukasten für hybride Apps und setzten dabei konsequent auf Webtechnologien, hochverfügbare und wiederverwendbare Cloud-Dienste sowie auf eine zunehmend Microservice-basierte Architektur. Durch Verwendung einheitlicher Rahmenvorgaben zur App-Entwicklung können Synergieeffekte zur Wiederverwendung von UI-Konzepten, Applikationsarchitekturen, Frameworks, Cloud-Diensten und Vorgehensweisen erreicht werden.

5 Evaluation und Diskussion

In den folgenden Abschnitten findet die Diskussion der vier Architekturvarianten und ihrer Implementierungen anhand von Nutzungsszenarien statt. Dabei werden zunächst die Kernbestandteile der jeweiligen Variante aufgezählt. Anschließend werden konkrete Systeme benannt, die nach der Architekturvariante umgesetzt wurden. Dann wird die Suche ausgehend von den Nutzungsszenarien bewertet und letztlich die Erkenntnisse daraus festgehalten.

5.1 Anwendungsszenarien (Use-Cases)

Die folgenden Anwendungsszenarien stellen typische Suchanfragen an ein Umwelt- bzw. Energieportal dar. Die Nutzer stammen aus der interessierten Öffentlichkeit (Szenarien 2, 3, 5) bzw. haben eine Bezug zur beruflichen Tätigkeit des Nutzers (Szenarien 1, 4), jedoch ohne tiefen fachlichen Bezug (Vorwissen, Expertenwissen) zur durchsuchten Domäne.

5.1.1 Szenario 1 „Politiker“

Der Lokalpolitiker Ernst Maier interessiert sich für den Ausbau mit erneuerbarer Energie in den Bereichen Windkraft und Photovoltaik, insbesondere den aktuellen Bestand und die damit erzeugte Leistung im Vergleich seines eigenen Landkreises Zollernalb mit den umliegenden Landkreisen.

Beispiele für mögliche Suchanfragen:

- „windkraft zollernalbkreis vergleich“
- „photovoltaik zollernalbkreis vergleich“
- „erneuerbare energie zollernalb“.

Erwartete Suchergebnisse:

- Windkraftanlagen und deren installierte Leistung im Zollernalbkreis und den umliegenden Landkreisen (Tübingen, Reutlingen, Sigmaringen, Tuttlingen, Rottweil und Freudenstadt), z.B. in MWh je Landkreis.
- Photovoltaikanlagen und deren installierte Leistung im Zollernalbkreis und den umliegenden Landkreisen (wie oben).
- Anteil der erneuerbaren Energien innerhalb der eigenen Kommune oder des eigenen Landkreises.

5.1.2 Szenario 2 „Bauen“

Das junge Ehepaar Silke und Jens Schmidt möchte im Konversionsgelände Karlsruhe-Knielingen einen Bauplatz kaufen, um darauf ein Haus zu bauen. Neben den Möglich-

keiten zum Einsatz erneuerbarer Energien und energieeffizientem Bauen interessieren sie sich für mögliche Umweltbelastungen am Standort, z.B. Altlasten, Lärm, Luftqualität.

Beispiele für mögliche Suchanfragen:

- „bauen in karlsruhe-knielingen“
- „energieeffizient bauen knielingen“
- „erneuerbare energie knielingen“
- „umwelt knielingen“
- „belastung konversionsgelände knielingen“.

Erwartete Suchergebnisse:

- Informationen über Baugebiete im Karlsruher Stadtteil Knielingen
- Informationen zum energieeffizienten Bauen (inkl. speziellen Informationen für das Baugebiet Karlsruhe-Knielingen)
- Informationen zu Möglichkeiten für den Einsatz erneuerbaren Energien (inkl. speziellen Informationen für das Baugebiet Karlsruhe-Knielingen)
- Beschreibung der Situation der Umwelt und möglicher Umweltbelastungen (Altlasten, Lärm, Luftqualität und weitere) rund um das Baugebiet Karlsruhe-Knieligen
- Weitere mögliche Standortfaktoren aus den Bereichen Umwelt und Energie.

5.1.3 Szenario 3 „Öko-Urlaub“

Die ökologisch und technisch sehr interessierte Familie Müller plant einen Urlaub im Nordschwarzwald. Bei Tagesausflügen möchten sie einige Beispiele für nachhaltige Energieerzeugung und -nutzung aufsuchen.

Beispiele für mögliche Suchanfragen:

- „ökologische energieerzeugung schwarzwald“
- „nachhaltige energie nordschwarzwald“
- „energiesparen schwarzwald“.

Erwartete Suchergebnisse:

- Fakten zur Nutzung erneuerbarer Energien im Schwarzwald, z.B. Konzepte und Projekte, z.B. Ökodörfer, Energieagenturen.
- Standorte mit Anlagen zur nachhaltigen Energieerzeugung (z.B. Windkraftanlagen, Solarparks, Wasserkraftanlagen, Stauseen, Biomasse) beschränkt auf den Nordschwarzwald
- Informationen zum Energiesparen im Schwarzwald, z.B. Möglichkeiten zur Gebäudedämmung, energieeffiziente Heizanlagen, Nutzung erneuerbarer Energien etc. sowie weiterführende Informationen, z.B. Energieagenturen im Schwarzwald

5.1.4 Szenario 4 „Solardächer“

Der Immobilienmakler Horst Bauer möchte die Exposés der von ihm angebotenen Objekte mit Informationen zur Eignung für thermische Solaranlagen bzw. Photovoltaikanlagen versehen. Er hat eine Liste mit Adressen von 20 bestehenden Einfamilienhäusern im Raum Heilbronn und sucht nach entsprechenden Daten.

Beispiele für mögliche Suchanfragen:

- „eignung solaranlage dach“
- „photovoltaik heilbronn“
- „solarenergie heilbronn schweinsbergstraße 12“
- „solar untergruppenbach habichthöhe 9“.

Erwartete Suchergebnisse:

- Allgemeine Informationen zu Solaranlagen und unterschiedlichen Typen von Dächern.
- Spezifische Informationen zur Eignung der Dachflächen von einzelnen Gebäuden in Heilbronn für die Bestückung mit thermischen Solaranlagen bzw. Photovoltaikanlagen.
- Spezifische Informationen zur Eignung der Dachflächen an einer spezifischen Adresse, z.B. in Heilbronn bzw. Untergruppenbach.

5.1.5 Szenario 5 „Ökostrom“

Der Student Kevin Sauber ist zur Aufnahme seines Bauingenieur-Studiums von Hamburg in die Karlsruher Oststadt gezogen. Für seine erste Studentenbude interessiert er sich für CO_2-neutralen Strom und entsprechende Tarife.

Beispiele für mögliche Suchanfragen:

- „karlsruhe oststadt ökostrom“
- „CO_2-neutraler strom karlsruhe“
- „ökotrom tarife karlsruhe oststadt“.

Erwartete Suchergebnisse:

- Grundlegende Informationen zu CO_2-neutralem Strom („Ökostrom“) in der Karlsruher Oststadt, Informationen zum Stromnetzbetreiber in der Karlsruher Oststadt
- Informationen zu Anbietern von Ökostrom-Tarifen (in Karlsruhe bzw. in der Karlsruher Oststadt)
- Konkrete Ökostrom-Tarife für den Standort Karlsruhe Oststadt.

5.2 Evaluation und Bewertung der ersten Architekturvariante

Die Kernbestandteile der Architektur sind:

- Nutzung von domänenspezifischen Vokabularen
    - Ergänzung der Volltextsuche um ein domänenspezifisches Wörterbuch
    - Automatische Suchworterweiterung (Vorschläge für Suchbegriffe)
- Anbindung externer Datenquellen über die Suchmaschine (OneBoxen)
- Darstellung von Suchergebnissen externer Datenquellen.

Die Evaluation erfolgt anhand der folgenden Systeme:

- Energieportal Baden-Württemberg; Volltextsuche; automatische Suchworterweiterung. Bis zu dessen Abschaltung im Einsatz.
- Landesumweltportale (LUPO); Domäne per Wörterbuch in der Volltextsuche. Seit 2007 bis heute im Einsatz.
- Landesumweltportale (LUPO); OneBoxen. Bis zur Ablösung durch serviceorientierte Architektur (gemäß dritter Architekturvariante) im Einsatz.
- Volltextsuche in „Meine Umwelt“. Seit 2007 bis heute im Einsatz.

Die Nutzungsszenarien sind nachfolgend zusammengestellt:

Quelle: Abfragen im Energieportal Baden-Württemberg

Szenario	Suchbegriffe	Volltext	Karte	Objekte	Bemerkung/Ziel
Politiker 1	windkraft zollernalbkreis vergleich	✓	◆	◆	
Politiker 2	photovoltaik zollernalbkreis vergleich	✓	◆	◆	
Politiker 3	erneuerbare energie zollernalb	✓	◆	◆	
Bauen 1	bauen in karlsruhe-knielingen	✓	◆	◆	
Bauen 2	energieeffizient bauen knielingen	✓	◆	◆	
Bauen 3	erneuerbare energie knielingen	✓	◆	◆	
Bauen 4	umwelt knielingen	✓	◆	◆	
Bauen 5	belastung konversionsgelände knielingen	✓	◆	◆	
Öko-Urlaub 1	ökologische energieerzeugung schwarzwald	✓	◆	◆	
Öko-Urlaub 2	nachhaltige energie nordschwarzwald	×	◆	◆	
Öko-Urlaub 3	energiesparen schwarzwald	✓	◆	◆	

Solardächer 1	eignung solaranlage dach	✓	◆	◆	
Solardächer 2	photovoltaik heilbronn	✓	◆	◆	
Solardächer 3	solarenergie heilbronn schweinsbergstraße 12	×	◆	◆	
Solardächer 4	solar untergruppen-bach habichthöhe 9	×	◆	◆	
Ökostrom 1	karlsruhe oststadt ökostrom	×	◆	◆	
Ökostrom 2	co2-neutraler strom karlsruhe	×	◆	◆	
Ökostrom 3	ökotrom tarife karlsru-he degenfeldstraße	×	◆	◆	

Volltext

✓ Gewünschte Information unter den ersten 20 Treffern.

× Gewünschte Information nicht unter den ersten 20 Treffern..

◆ = nicht anwendbar

Das Energieportal bietet neben einem thematischen Zugang mit Verweisen zu spezifischen Themen eine Volltextsuche an, deren Wörterbuch inhaltlich um das Thema „Energie“ aus dem GEMET-Thesaurus (European Environment Information and Observation Network 2017) erweitert ist. Die Volltextsuche liefert in den meiste Fällen brauchbare Ergebnisse, sie liefert jedoch grundsätzlich keine konkreten Datenobjekte und keine Geodaten, daher sind die Spalten „Karte“ und „Objekte“ auf die erste Architekturvariante nicht anwendbar.

In einigen Fällen werden keine passenden Ergebnisse gefunden:

- Ortssuche zu spezifisch (Solardächer 3+4, Ökostrom 3), z.B. Straßennamen, die im Volltext nicht verfügbar sind
- Thema ist nicht in den Datenquellen (hier Volltextsuche) enthalten: Ökostrom 1,2
- Ortsangabe „Nordschwarzwald“ liefert in Kombination mit den gesuchten Themen keine Treffer (Öko-Urlaub 2).

Im Energieportal stehen keine Datenquellen für Objekte per OneBox zur Verfügung. Dagegen hat sich der Mechanismus im Umweltportal Baden-Württemberg bewährt, z.B. zum Auffinden von Messwerten (Luftqualität, Pegel). Ein großes Manko ist allerdings der Mechanismus zur Abfrage, der immer auf den Suchbegriffen der Volltextsuche basiert. Die verwendete Volltextsuchmaschine Google Search Appliance erlaubt keine Auflösung von Geobegriffen.

Bewertung und Erkenntnisse

Die Verbesserung beschränkt sich im Energieportal auf die Volltextsuche – hier allerdings ohne erkennbaren Mehrwert für die betrachteten Szenarien. Im Umweltportal gibt es dagegen eine ganze Reihe von Verbesserungen im Bereich der Abbildung von Umgangssprache auf Fachsprache, z.B. „Müll“ suchen, Informationen zum Thema „Abfall“ finden.

Die Semantik der Zielsysteme (hier: Websites) wird nicht erfasst. Es werden lediglich domänenspezifische Begriffe, z.B. über Synonymketten innerhalb der Volltextsuche aufgelöst. Eine Abbildung zwischen Umgangssprache und Fachbegriffen findet zwar statt, ist allerdings im Bereich Energie weniger hilfreich als im Umweltbereich.

Die Anzeige von Treffern beschränkt sich auf die Volltextsuche und OneBoxen. Es werden keine Karten und keine (Energie-)Objekte gefunden, da keine Datenquellen zum Thema Energie per OneBox (Google Inc. 2015) an die Suchmaschine angebunden sind. Trigger für OneBoxen können nur die verwendeten Suchbegriffe sein, nicht jedoch z.B. explizite Geokoordinaten oder Datum/Zeitangaben.

Der Anschluss von Zielsystemen erfolgt (im Falle des Umweltportals) jeweils über Adapter, da OneBoxen keine standardisierte Schnittstelle darstellen. Der direkte Anschluss von langsamen Zielsystemen als OneBox funktioniert wegen des vorgegebenen Timeouts nur über Caching, Indexbildung oder ähnlichen Techniken, d.h. durch redundante Datenbereitstellung.

Die Verbesserungen durch die erste Architekturvariante sind (erwartungsgemäß) beschränkt.

Dennoch lassen sich einige wertvolle Erkenntnisse daraus ziehen:

- Die Verwendung domänenspezifischer Vokabulare ist für eine semantische Suche unbedingt notwendig. Mehrere Vokabulare können allerdings bei übergreifenden Themenfeldern fragmentiert sein. Ein Zusammenführen der verschiedenen Vokabulare ist notwendig.
- Domänenspezifische Vokabulare können als Erweiterung des Wörterbuches von suchmaschineneigenen Verbesserungen der Suche (Synonymketten), z.B. zur Auflösung von umgangssprachlichen Begriffen dienen.
- Suchbegriffe alleine reichen für die sinnvolle Anfrage vieler Zielsysteme nicht aus. Es ist notwendig, die Semantik der Suchbegriffe zu verstehen, d.h. sie einem Wortgut (Vokabular) zuzuordnen, oder sie anderweitig, z.B. als Ortsbegriffe, zu klassifizieren. Zusätzlich ist ggf. eine Anreicherung bzw. Explizierung der Suchbegriffe notwendig, z.B. um aus Ortsnamen Geokoordinaten zu gewinnen.
- Zielsysteme müssen passende Schnittstellen bereitstellen. Die Daten in den Zielsystemen müssen semantisch beschrieben werden, minimal als Abbildung ihrer Konzepte auf eines der verwendeten Vokabulare.
- Nicht verfügbar gemachte Zielsysteme ergeben Lücken in den Trefferlisten, es wird nicht das gesamte Potenzial der Suche ausgenutzt. Eine halbherzige Umsetzung (Anbindung nur weniger Zielsysteme aus einer größeren Auswahl) schafft Intransparenz und letztendlich Frust beim Nutzer.

5.3 Evaluation und Bewertung der zweiten Architekturvariante

Die Kernbestandteile der Architektur sind:

- Serverseitige Verarbeitung von Anfragen, Abfrage von Ergebnissen per SearchBroker und Erzeugung von Trefferansichten
    - Beschreibung von Zielsystemen (externe Datenquellen) per OpenSearch-Description
    - Vorverarbeitung und Anreicherung von Suchbegriffen, z.B. mithilfe von Gazetteer-Services
    - Thematische Auflösung von Suchbegriffen über ein Ontologiesystem
    - Anfrage von Zielsystemen
    - Erzeugung eines Ergebnismashups (Trefferansicht)
    - Mashup-Komponenten für verschiedene Datentypen.
- Nutzung von domänenspezifischen Vokabularen
    - Mehrere Vokabulare innerhalb eines Ontologiesystems
    - Vernetzung von Teilontologien durch Artikulationsontologie.

Die Evaluation erfolgt anhand der folgenden Systeme:

- Landesumweltportale (LUPO), WebGenesis-basierte Version.
- Semantische Suche innerhalb des SUI-Projektes. Die semantische Suche aus SUI kam über den Prototyp-Status hinaus nie in den produktiven Einsatz.

Als Nutzungsszenarien werden wieder die bereits für die erste Architekturvariante betrachteten Szenarien verwendet:

Quelle: Umweltportal Baden-Württemberg (Prototyp, Webgenesis-basierte Version), SUI-Suche

Szenario	Suchbegriffe	Volltext	Karte	Objekte	Bemerkung/Ziel
Politiker 1	windkraft zollernalbkreis vergleich	✓	✓	✓	Kein „Vergleich“
Politiker 2	photovoltaik zollernalbkreis vergleich	✓	✓	✓	Kein „Vergleich“
Politiker 3	erneuerbare energie zollernalb	✓	✓	✓	
Bauen 1	bauen in karlsruhe-knielingen	✓	✓	×	Lebenslage „Bauen“ Karte „Lärm“
Bauen 2	energieeffizient bauen knielingen	✓	✓	×	Lebenslage „Bauen“ Karte „Lärm“
Bauen 3	erneuerbare energie knielingen	✓	✓	×	

Bauen 4	umwelt knielingen	✓	✓	✓	Viele Karten, z.B. Schutzgebiete
Bauen 5	belastung konversionsgelände knielingen	✓	×	×	
Öko-Urlaub 1	ökologische energieerzeugung schwarzwald	✓	✓	✓	
Öko-Urlaub 2	nachhaltige energie nordschwarzwald	×	✓	✓	
Öko-Urlaub 3	energiesparen schwarzwald	✓	×	×	
Solardächer 1	eignung solaranlage dach	✓	✓	×	
Solardächer 2	photovoltaik heilbronn	✓	✓	×	
Solardächer 3	solarenergie heilbronn schweinsbergstraße 12	×	✓	×	Mapping Adresse fehlt in Daten
Solardächer 4	solar untergruppenbach habichthöhe 9	×	✓	×	Mapping Adresse fehlt in Daten
Ökostrom 1	karlsruhe oststadt ökostrom	×	×	×	Fehlende Daten
Ökostrom 2	co2-neutraler strom karlsruhe	×	×	×	Fehlende Daten
Ökostrom 3	ökotrom tarife karlsruhe degenfeldstraße	×	×	×	Fehlende Daten

Volltext
✓ Gewünschte Information unter den ersten 20 Treffern.
× Gewünschte Information nicht unter den ersten 20 Treffern.

Karte
✓ Gewünschte Information auf Kartenansicht enthalten.
✓ Gewünschte Information auf Kartenansicht enthalten, Ortsauswahl korrekt
× Gewünschte Information nicht auf Kartenansicht enthalten.

Objekt
✓ Gewünschte Information als Objekt enthalten.
× Gewünschte Information nicht als Objekt enthalten.

◆ = nicht anwendbar

orange = keine Datenquelle verfügbar

Die Volltextsuche (spalte „Volltext“) liefert praktisch die gleichen Ergebnisse wie in der ersten Architekturvariante, sie basiert auf derselben Suchmaschine unter Verwendung des erweiterten Wörterbuchs, in dem jedoch nicht alle Inhalte des Ontologiesystems (Lebenslagen, Artikulationsontologie) verfügbar sind.

Eine deutliche Verbesserung bei der Ansprache von Karten und Objekten bietet die Lebenslagenontologie, da zu allgemeinen Begriffen wie „Bauen“ spezifische Themen (z.B. „Lärm“, „Altlasten“, „Schutzgebiete“ etc.) gefunden werden, die zur konkreten Anfrage von Zielsystemen dienen können. Der Mechanismus funktioniert, Zielsysteme werden erkannt und Anfragen können gestellt werden. Im Vergleich mit der ersten Architekturvariante stehen in vier der 5 Szenarien zusätzlich Geoinformationen auf Karten zur Verfügung, für die Szenarien „Politiker“, „Bauen 4“ und „Öko-Urlaub“ zusätzlich auch konkrete Objektinformationen.

Die Verwendung einer Ortsontologie auf Basis einer großen Geodatenbasis (hier: Semantic Network Service) ermöglicht in den meisten Fällen das Erkennen des Ortsbezugs und die entsprechende Fokussierung des Kartenausschnittes.

Die Szenarien „Politiker 1“ und „Politiker 2“ mit ihrem Schlüsselwort „vergleich“ zeigen, dass die inhaltliche Analyse der Suchanfrage auf Basis der verwendeten Ontologie mit ihren Teilen GEMET, Lebenslagen, Orte, und Artikulation bereits für die vorliegenden Szenarien nicht vollständig ist, und weitere inhaltliche Bestandteile in die Ontologie aufgenommen werden müssen. Der Benutzer erwartet einen Vergleich, d.h. eine Gegenüberstellung, der Eigenschaften verschiedener Windkraftanlagen des Zollernalbkreises. Eine ähnliche Erwartung, zumindest an die Präsentation mehrerer passender Ergebnisse, impliziert das Schlüsselwort „Tarife“ (im Plural) in Szenario „Ökostrom 3“. „Tarife“ ist jedoch in keiner der Teilontologien enthalten.

Bewertung und Erkenntnisse

Das Erkennen der Semantik der Suchanfrage ist zur gezielten Anfrage konkreter Zielsysteme, deren Semantik in Form von Zielsystembeschreibungen ebenfalls bekannt ist, sinnvoll. Neben einem oder mehreren Vokabularen, die möglichst die ganze Domäne abdecken sollten, können weitere Vokabulare, z.B. Lebenslagen und Ortsbezeichnungen, verwendet werden. Insbesondere die Einbindung der Lebenslagen bieten dem Nutzer im Grundsatz ein gutes Bild zu konkreten Fragestellungen – sofern die notwendigen Daten verfügbar sind. Die Vokabulare können in Form von Ontologien dargestellt werden, einfacher strukturierte Vokabulare, die z.B. in Form von Thesauri oder Taxonomien vorliegen, lassen sich in solche transformieren. Einzelne Vokabulare (Teilontologien) können mithilfe einer Artikulationsontologie miteinander verknüpft werden. Der Aufwand für die Erstellung der Artikulationsontologie wächst jedoch mit der Anzahl der vorhandenen Teilontologien mindestens quadratisch, in der Praxis muss deren Anzahl daher beschränkt bleiben. Auch wenn die Verknüpfungen zwischen den Teilontologien sich teilweise generieren lassen - sie müssen jedoch zumindest in Teilen manuell bearbeitet werden. Allgemeine Konzepte wie Orte oder Zeitangaben lassen sich daneben über spezialisierte Dienste (Gazetteer-Dienste) erkennen, explizieren und anreichern.

Das Einfügen konkreter Objekte aus den Zielsystemen in die Ontologie und die Verknüpfung der Objekte mit Konzepten aus den Domänenontologien haben sich nicht bewährt, da die verwendeten Ontologiesysteme mit einer großen Zahl enthaltener Instanzen erheblich an Performanz (inakzeptable Antwortzeiten) einbüßen. Daneben

muss dauerhaft die Konsistenz zwischen den Daten in den Zielsystemen und denen in der Ontologie sichergestellt werden. Daher, und da in der zweiten Architekturvariante die Daten grundsätzlich aus den Zielsystemen abgerufen werden sollten, wurde die Integration der Daten in die Ontologie nicht weiter verfolgt.

Die verwendeten Zielsystembeschreibungen zum parametrisierten Zugriff auf Zielsysteme funktionieren grundsätzlich, teilweise realisiert über Adapter, falls die Zielsysteme nicht über URLs adressierbar sind, jedoch lassen sich nicht alle gewünschten Zielsysteme mangels technischer Schnittstellen tatsächlich nutzen. Das schränkt die Suche beträchtlich ein, da viele gefundene Themen und Konzepte nicht durch entsprechende Zielsysteme mit den entsprechenden Daten (und damit Suchtreffern) bedient werden können. Daher laufen viele Suchanfragen ins Leere oder bieten nur Fragmente einer sinnvollen Antwort. Eine möglichst vollständige Abdeckung der Domäne ist daher wünschenswert, hängt jedoch von der Verfügbarkeit der Informationen für die Suche ab. Daher sollte der Zugang zu den Daten vor der Implementierung einer konkreten Suchmaschine geprüft werden.

Das grundsätzliche Funktionieren der Zielsystembeschreibungen für den parametrisierten Zugriff auf Zielsysteme demonstriert in der Praxis neben dem Prototypen der SUI-Suche die mobile App „Meine Umwelt“ (Schlachter et al. 2013), innerhalb der alle enthaltenen Datenquellen über erweiterte OpenSearch-Descriptions beschrieben werden.

Der SearchBroker als koordinierendes und prozessierendes Element zwischen Suchanfrage, deren semantische Erkennung und Anreicherung und der Anfrage von Zielsystemen funktioniert ebenfalls grundsätzlich.

Das prototypische System zur semantischen Suche innerhalb des Umweltportals Baden-Württemberg zeigt insgesamt zwei erhebliche Einschränkungen:

1. Die serverseitige Abfrage von Zielsystemen bildet einen Flaschenhals. Es muss auf „langsame“ Zielsysteme, die teilweise erst mit erheblichen Latenzen von mehreren Sekunden antworten, warten. Das bedeutet, dass auch die Antwortzeit der Suche entweder durch Timeouts in den Anfragen an die Zielsysteme begrenzt – dafür jedoch auf mögliche Ergebnisse verzichtet – werden muss, oder dass die Suche immer so lange läuft, bis alle Zielsysteme Daten (oder zumindest eine ggf. negative Antwort) geliefert haben, was zu erheblichen, inakzeptablen Wartezeiten[21] für den Nutzer führt.
2. Viele mögliche Zielsysteme bieten keine Schnittstellen, die sich zur maschinellen Abfrage eignen, sondern lediglich Weboberflächen für menschliche Nutzer. Sie sind als Zielsysteme für die zweite Architekturvariante ungeeignet. Da auch

[21] Für die Landesumweltportale wurde festgelegt, dass die Suchmaschine nach spätestens 1000ms ein Ergebnis ausliefern muss. Experimente von Google (Brutlag 2009) und Microsoft/Bing (Schurman und Brutlag 2009) zeigten, dass bei Internet-Suchmaschinen bereits Wartezeiten von einigen 100ms zu einer messbaren Abnahme der Anzahl der Suchen je Tag führen.

die zentrale Rechercheplattform für Umwelt- und Energieinformationen der LUBW, Umweltdaten und -karten online (UDO) (LUBW 2016), die auf dem Produkt Cadenza Web (disy Informationssysteme GmbH 2016) basiert, davon betroffen ist, bleiben sehr große Datenbestände (z.B. Objekte, Zeitreihen) für die Suche unerschlossen. Eine Öffnung von UDO über eine entsprechende Schnittstelle konnte durch die Herstellerfirma während der Laufzeit des SUI-Projektes von 2009 bis 2012 nicht realisiert werden, da ein solches Vorgehen nicht der Strategie der Herstellerfirma entsprach (Bügel et al. 2011b), und die hierfür notwendigen Aufwände aus dem laufenden Forschungsprojekt nicht getragen werden konnten. Die wertvollsten Datenschätze der LUBW blieben daher für die Suche unerschlossen.

Wenn entscheidende Systeme einen derart massiven Einfluss auf die Gesamtarchitektur haben können, muss das Paradigma, Daten direkt, d.h. zur Laufzeit der Anfrage, aus den Zielsystemen abzurufen, grundsätzlich infrage gestellt werden. Wenn der direkte Zugriff auf die Zielsysteme, z.B. mangels technischer Schnittstellen oder aus Gründen der Performanz, also nicht möglich ist, sollten die Daten über redundante Systeme bereitgestellt werden, die entsprechende Schnittstellen, Verfügbarkeit und Performanz bieten.

Hinzu kommt, dass die serverseitige Erzeugung des Ergebnis-Mashups auf Basis von Templates die Entwicklung und Anpassung dynamischer Trefferlisten erschwert, da für Änderungen jeweils serverseitige Anpassungen notwendig sind, die nicht über die Redakteur-/Nutzeroberfläche vorgenommen werden können. Das ist vor allem eine Einschränkung, die das (durch Projektvorgaben) verwendete Content Management System Webgenesis, Version 7.13, (Fraunhofer IOSB 2016) betrifft, bei dem Templates im Dateisystem, nicht in der Datenbank vorgehalten und nur einmalig beim Start des Systems eingelesen werden. Im Allgemeinen lässt sich das Problem durch Nutzung entsprechender dynamischer Portal- oder CMS-Systeme jedoch lösen.

Die zweite Architekturvariante verbessert die Suche gegenüber der ersten im Ergebnis deutlich, insbesondere durch das Spektrum möglicher Zielsysteme und der darin enthaltenen Daten, die semantische Verarbeitung der Suchanfrage und die vielfältigere Ergebnispräsentation. Es gibt jedoch deutliche Einschränkungen, die vor allem durch die ausschließlich serverseitige Prozessierung der Suche (v.a. SearchBroker, Zielsystem-Anfragen, Ergebnis-Mashup) ausgelöst werden. Hinzu kommt die mangelnde Verfügbarkeit von Schnittstellen zu wichtigen datenhaltenden Systemen, die sich im Sinne der Architektur so nicht als Zielsysteme eignen. Daher wurde die Suche nach der zweiten Architekturvariante nicht produktiv eingesetzt, es lassen sich dennoch wertvolle Erkenntnisse aus den Entwicklungen und Erfahrungen ziehen, die in weitere Architekturvarianten eingeflossen sind:

- Ontologiesysteme sind zur Darstellung und Verknüpfung mehrere Vokabulare mithilfe einer Artikulationsontologie grundsätzlich geeignet.
- Die Pflege der (Teil-)Ontologien und Bereitstellung der Artikulationsontologie macht Aufwand; das Hinzufügen weiterer Teilontologien vermehrt den Aufwand

sogar überproportional stark. Daher sollte hier soweit wie möglich automatisiert werden.

- Die rein serverseitige Implementierung der Suche funktioniert insbesondere wegen der synchronen Anfragen der Zielsysteme nicht ausreichend performant, auch wenn die einzelnen Teilkomponenten der Suche im Grundsatz funktionieren, sowohl einzeln als auch im Zusammenspiel.
  Mögliche Ansatzpunkte hierfür sind die Verlagerung der Kernfunktionalitäten (semantische Verarbeitung der Suchanfrage, Koordination durch den Search-Broker, Anfrage der Zielsysteme, Aufbereitung der Trefferansicht (Mashup)) in den Client, insbesondere durch Nutzung asynchroner Aufrufe von Hintergrunddiensten und Zielsystemen.
- Bereitstellung von Daten aus Systemen, die nicht direkt als Zielsysteme geeignet sind, über redundante Systeme mit geeigneten Schnittstellen. Hier müssen die Konsistenz bzw. Kohärenz der Originaldaten und der Daten im Redundanzsystem sichergestellt werden, z.B. durch regelmäßige Synchronisation oder durch entsprechende Protokolle.

5.4 Evaluation und Bewertung der dritten Architekturvariante

Die Kernbestandteile der Architektur sind:

- (Redundante) Bereitstellung von Daten über eine Reihe generischer Datendienste (Webcache)
- Transformation und Synchronisation der Daten zwischen Zielsystemen und Webcache mithilfe von definierten Prozessen (Data Ingestion)
- Nutzung von Vokabularen zur semantischen Beschreibung der Daten (bzw. von Klassen); Beschreibung der Daten durch Schemata
- Prozessierung und Koordination der Suchanfrage und der Anfrage von Zielsystemen im Client mithilfe spezialisierter Gazeteer-Dienste
- Ergebnispräsentation durch eine Reihe generischer Frontend-Komponenten
- Orchestrierung der Suche im Client und der Ergebnispräsentation durch Event-basierte Kommunikation zwischen den Frontend-Komponenten.

Die Evaluation erfolgt anhand der folgenden Systeme:

- Landesumweltportale (LUPO), Liferay-basierte Version der Länder Baden-Württemberg, Sachsen-Anhalt und Bayern.
- Energieatlas Baden-Württemberg.

Als Nutzungsszenarien werden erneut die zuvor betrachteten verwendet:

Quelle: Umweltportal Baden-Württemberg 2015, Liferay-basierte Version

Szenario	Suchbegriffe	Volltext	Karte	Objekte	Bemerkung/Ziel
Politiker 1	windkraft zollernalb-kreis vergleich	✓	✓	✓	Kein Vergleich
Politiker 2	photovoltaik zollernalb-kreis vergleich	✓	✓	✓	Kein Vergleich
Politiker 3	erneuerbare energie zollernalb	✓	✓	✓	
Bauen 1	bauen in karlsruhe-knielingen	✓	✓	× (✓)	Lebenslagen („Bauen“) nicht instrumentiert
Bauen 2	energieeffizient bauen knielingen	✓	✓	× (✓)	Lebenslagen („Bauen“) nicht instrumentiert
Bauen 3	erneuerbare energie knielingen	✓	✓	×	Lebenslagen (z.B. „Planen der Heizungsanla-ge“) nicht instru-mentiert
Bauen 4	umwelt knielingen	✓	✓	✓	Viele Karten, z.B. Schutzge-biete
Bauen 5	belastung konversi-onsgelände knielingen	✓	×	×	
Öko-Urlaub 1	ökologische energieer-zeugung schwarzwald	✓	✓	✓	
Öko-Urlaub 2	nachhaltige energie nordschwarzwald	×	✓	✓	
Öko-Urlaub 3	energiesparen schwarzwald	✓	×	× (✓)	Lebenslage
Solardächer 1	eignung solaranlage dach	✓	✓	✓	
Solardächer 2	photovoltaik heilbronn	✓	✓	✓	
Solardächer 3	solarenergie heilbronn schweinsbergstraße 12	×	✓	×	Mapping der Adresse fehlt in den Daten
Solardächer 4	solar untergruppen-bach habichthöhe 9	×	✓	×	Mapping der Adresse fehlt in den Daten

Ökostrom 1	karlsruhe oststadt ökostrom	×	×	×	Fehlende Daten
Ökostrom 2	co2-neutraler strom karlsruhe	×	×	×	Fehlende Daten
Ökostrom 3	ökotrom tarife karlsruhe degenfeldstraße	×	×	×	Fehlende Daten

Volltext
✓ Gewünschte Information unter den ersten 20 Treffern.
× Gewünschte Information nicht unter den ersten 20 Treffern.

Karte
✓ Gewünschte Information auf Kartenansicht enthalten.
✓ Gewünschte Information auf Kartenansicht enthalten, Ortsauswahl korrekt
× Gewünschte Information nicht auf Kartenansicht enthalten.

Objekt
✓ Gewünschte Information als Objekt enthalten.
× Gewünschte Information nicht als Objekt enthalten.

◆ = nicht anwendbar

orange = keine Datenquelle verfügbar

Die Suchergebnisse der dritten Architekturvariante unterscheiden sich von denen der vorigen beiden Varianten vor allem durch die bessere Verfügbarkeit konkreter Objektdaten in der Trefferliste. Grund dafür ist hauptsächlich die Verwendung des Webcache, der Daten aus erheblich mehr Zielsystemen für die Suche verfügbar macht. Dennoch gibt es auch hier Szenarien (v.a. „Ökostrom“, „Solardächer 3 und 4“), die keine konkreten Treffer liefern. Eine Ursache hierfür liegt in den Daten, z.B. sind die für solare Nutzung geeigneten Dachflächen nicht adressscharf verfügbar. Dennoch kann der Nutzer sich im entsprechenden Kartenlayer, der zumindest ortsscharf angezeigt wird, orientieren und die gewünschten Dachflächen finden. Die adressscharfe Ortsbestimmung über den verwendeten Gazetteer-Dienst ist zwar grundsätzlich verfügbar, ist jedoch bewusst deaktiviert, da nur wenige Datensätze über Adressen gefunden werden können bzw. Adressen aus Datenschutzgründen im Webcache nicht enthalten sind. Dennoch bewährt sich auch hier die Anreicherung der Ortsinformationen durch Geokoordinaten durch den Geo-Gazetteer-Dienst, denn sie stellen einen universellen Zugang bei der Suche nach Objekten mit Ortsbezug dar.

Im Szenario „Bauen“ sind die im Umweltportal gelieferten Suchergebnisse tatsächlich sogar schlechter als in Architekturvariante zwei, in der die Lebenslage „Bauen“ in die Ontologie integriert ist, und daher eine ganze Reihe von konkreten Themen in die Suche einbezieht. In der hier verwendeten Version des Umweltportals sind alle thematischen Bezüge ebenfalls über Konzepte aus der Domänenontologie beschrieben. Aus Performanzgründen wird die thematische Zugehörigkeit von Datensätzen zu Konzepten der Domäne bereits beim Transferieren der Daten in den Webcache festgelegt und dort gespeichert, teilweise durch Reduzierung der Konzepte auf ihre Namen (Labels), um sie besser indexieren zu können. Dabei kommt zwar dasselbe Ontologiesystem wie

in Architekturvariante zwei zum Einsatz, d.h. die berechnete „Umgebung“ eines Konzeptes (z.B. „Naturschutzgebiet“) erfolgt nach demselben Prinzip wie sie die dortige thematische Zuordnung bei der Verarbeitung der Suchanfrage liefert, allerdings wurde die Ontologie wegen des Pflegeaufwands gegenüber dem prototypischen SUI-System auf die Inhalte des GEMET reduziert, weshalb in der Suche des produktiven Systems keine Lebenslagen verfügbar sind.

Bewertung und Erkenntnisse

Der Webcache mit seinen (Micro-)Services ist sehr gut geeignet, um den Zugriff auf Daten und Objekte von ihren jeweiligen Zielsystemen zu entkoppeln und generell die Verfügbarkeit von Daten zu gewährleisten, weitgehend ohne die Zielsysteme zu belasten. Das Aufsetzen und der Betrieb des Webcache sowie die Synchronisation der Daten aus den Zielsystemen mit dem Webcache bedeuten zwar zusätzlichen Aufwand, der sich nach der ersten Einrichtung jedoch sehr gut automatisieren lässt.

Das modulare Konzept von auf bestimmte Datentypen spezialisierten, dennoch generischen Datendiensten hat sich bewährt, die acht Haupt- und zwei Hilfsdienste des Microservice-Backends decken den Bedarf für alle im Energie- und Umweltbereich anfallenden Datentypen wie Stammdaten, Zeitreihen, Geodaten, digitalen Assets etc. ab. Die einzelnen Dienste sind durch ihre Konzeption und Implementierung als Microservices relativ unkompliziert und bieten einfache REST-Schnittstellen zur Nutzung, die über Versionierung dauerhaft stabil gehalten werden können. Durch Verwendung skalierbarer Backend-Systeme und Laufzeitinfrastrukturen können sie ausreichend Leistung für unterschiedliche Nutzungsszenarien bieten. Die Nutzung von Containervirtualisierung mittels Docker bietet Flexibilität beim Betrieb und ebenfalls Vorteile bei der Skalierung in entsprechenden Infrastrukturen wie Kubernetes (Kubernetes 2016).

Da der Webcache wenige, stabile Schnittstellen anbietet, reduziert sich der Aufwand für die Implementierung Schnittstellen im Client erheblich, was die Implementierung von Frontend-Komponenten wesentlich vereinfacht – die Schnittstellen zu den Zielsystemen müssen allerdings an anderer Stelle, bei der Datentransformation, implementiert werden.

Das Sucherlebnis für die Nutzer unterscheidet sich gegenüber der Architekturvariante zwei erheblich. Die Suchseite besteht aus einem Zusammenspiel verschiedener Komponenten im Client, die über eine Event-basierte Kommunikationsschicht miteinander verbunden sind und darüber orchestriert werden. Die einzelnen Komponenten werden so mit der rohen Suchanfrage, mit ihrer semantisch verarbeiteten Form sowie den Ergebnissen der verschiedenen Zusatzdienste (Gazetteers) versorgt. Sie können so autark den Webcache anfragen und die gefunden Daten zur Anzeige bringen. Die Kommunikation mit den Hintergrunddiensten geschieht asynchron, so dass der Nutzer nicht mehr warten muss, bis das Gesamtergebnis vorliegt, sondern meist sehr schnell mit ersten Ergebnissen, z.B. der Trefferliste der Volltextsuchmaschine, versorgt wird, während aufwändigere Komponenten, wie z.B. die Kartenansicht, erst nach und nach die vollständigen Ergebnisse, z.B. die zum Thema passenden Kartenlayer, anzeigen.

Die per REST-URLs verfügbaren Daten bieten bereits eine gute Grundlage für die Nutzung als Linked Data im Sinne des Semantic Web, jedoch fehlen den Diensten derzeit noch die dafür notwendigen standardisierten Datenformate, z.B. RDF. Zusätzlich sollen in den Zielsystemen nicht enthaltene Beziehungen zwischen den Daten darstellbar sein, die jedoch erst erzeugt werden müssen, z.B. unter Ausnutzung von Zusatzwissen. Der in der Architektur beschriebene Link-Service wird in der Praxis der Umweltportale noch nicht genutzt, unter anderem, da für die Beschreibung und Konfiguration von Verknüpfungen und den notwendigen Implementierungen bei den Transformationen Aufwände anfallen, die beim erstmaligen Aufbau des Webcache nach der dritten Architekturvariante zunächst nicht leistbar waren.

5.5 Diskussion der vierten Architekturvariante

Die Kernbestandteile der Architektur sind:

- Zusätzlicher Link-Service
- Erweiterung der Datentransformation in den Webcache um die Erzeugung zusätzlicher Verknüpfungen
- Erweiterung der Anzeigekomponenten um die Darstellung zusätzlicher, verknüpfter Informationen.

Derzeit erfolgte noch keine Umsetzung in konkreten Systemen. Eine Evaluation bezüglich der konkreten Nutzungsszenarien hat daher noch nicht stattgefunden.

Diskussion

Eine abschließende Bewertung der vierten Architekturvariante ist zum Zeitpunkt der Erstellung der vorliegenden Arbeit noch nicht möglich. Die Fortführung der Arbeiten ist Bestandteil einer weiteren Dissertation. Aus der prototypischen Umsetzung liegen jedoch bereits einige Erkenntnisse vor.

Ein häufiges Problem stellt die Aggregation von Daten auf verschiedenen Ebenen dar. Sie lässt sich häufig nur unter Einbeziehung von Zusatzwissen bzw. durch die Verknüpfung verschiedener Datenquellen oder Datensätze erreichen. Zum Beispiel liegen Geoinformationen meist in Form von Koordinaten eines Objektes (z.B. als Punkt oder Polygonzug) vor. Um die Daten auf verschiedene Aggregationsstufen, z.B. Verwaltungseinheiten wie Gemeinde, Landkreis, Regierungsbezirk oder Bundesland, abzubilden, können zunächst die (vorliegenden) Strukturen, z.B. „Gemeinde G liegt in Landkreis L, Landkreis L liegt in Regierungsbezirk R, Regierungsbezirk R liegt in Bundesland B“, als Verknüpfungen im Linkservice abgelegt werden. Nun können, z.B. mithilfe eines Reverse-Geo-Gazetter-Dienstes, der z.B. Geokoordinaten auf eine Gemeinde abbilden kann, die Verknüpfung zwischen Objekten, z.B. Windkraftanlagen, mit den Gemeinden ebenfalls im Link-Service gespeichert werden. Durch Einsatz entsprechender Inferenztools/Reasoner lassen sich aus der Verwaltungsstruktur und der Verknüpfung zwischen Gemeinden und den dort befindlichen Windkraftanlagen die Anzahl der Windkraftanlagen (und ggf. deren Eigenschaften wie die maximale Leistung) leicht

auf den verschiedenen Verwaltungsebenen aggregieren, ohne dabei erneut Geooperationen ausführen zu müssen. Für die prototypische Umsetzung wurde für die Darstellung der Beziehungen kein Triplestore, sondern eine neo4j-Graphdatenbank (neo4j 2016) verwendet, die allerdings noch nicht an den eigentlichen Linkservice angebunden wurde. Sie demonstriert jedoch eindrucksvoll das Potenzial der generierten Verknüpfungen, da Attribute für die verschiedenen Verwaltungsebenen aggregiert werden können. Die in der Graphdatenbank vorhandenen Verwaltungsstrukturen lassen sich so mit beliebigen Datensätzen verknüpfen und in analoger Weise nutzen.

Ebenfalls ein großes Potenzial besteht bei der Verknüpfung von digitalen Textdokumenten, z.B.PDF- oder Office-Dokumenten zu Themen/Konzepten aus den Vokabularen der Domäne(n). Die automatische Klassifikation von Dokumenten des Umweltbereichs ist beispielsweise über den existierenden Semantic Network Service des Umweltbundesamtes (Umweltbundesamt 2016) möglich. Damit lassen sich direkt thematische, örtliche und zeitliche Einordnungen von Dokumenten automatisiert erstellen und entweder zusätzlich zu den Metadaten des Dokumentes oder in Form zusätzlicher Verknüpfungen innerhalb des Link-Services speichern und abfragen. Somit können z.B. Listen relevanter Dokumente zu einem Thema oder einer Auswahl von Themen leicht erzeugt werden.

Da zur Adressierung von Datensätzen die URLs der Datendienste verwendet werden, bietet das Gesamtszenario das Potenzial für die Verwendung als Linked Data. Dazu müssen die Datendienste lediglich die hierfür notwendigen Datenformate, d.h. RDF bzw. RDF/JSON, generieren, was für den Stammdatendienst bereits prototypisch erfolgt ist und damit die Erfüllung aller vier Kriterien für Linked Data:

1. Verwendung von URIs zum Benennen von Objekten
2. Verwendung von HTTP-URIs zum tatsächlichen Auffinden von Objekten
3. Verwendung von Standards wie RDF und SPARQL
4. Bereitstellung von Links in Form von URLs, um Verbindungen zu weiteren Objekten finden zu können.

Eine praktische Erweiterung ist die Möglichkeit zur erweiterten Abfrage von Daten aus allen Diensten, d.h. Anfragen, die alle oder bestimmte Verknüpfungen eines Objektes mit anderen Objekten enthält, auch wenn sie nicht im konkreten Datendienst, sondern nur im Link-Service verfügbar sind. Hierfür steht der Messaging-Kanal der Microservice-basierten Architektur zur Verfügung. Hierüber kann der Datendienst sich die notwendigen Informationen vom Link-Service holen und direkt in die Antwort einfügen. Die entsprechende Funktion muss jedoch noch umgesetzt werden.

6 Zusammenfassung

Das Ziel der vorliegenden Dissertationsschrift bestand darin, ein neues Konzept für die semantische Suche in heterogenen Informationssystemen zu Fragestellungen aus Umwelt und Energie zu entwickeln, d.h. die Konzeption und Entwicklung einer Suchfunktion für Webportale, die zwar für den Nutzer so einfach wie herkömmliche Internet-Suchmaschinen funktioniert, jedoch qualitativ bessere, ggf. mehr Ergebnisse liefert als eine konventionelle Volltextsuche. Dabei sollten folgende wissenschaftlichen Teilziele erreicht werden:

- Erkennen der Semantik einer Suchanfrage innerhalb einer gegebenen Domäne
- Beschreibung der Semantik von Daten gegebener Informationssysteme bezüglich einer vorgegeben Domäne, die sich ggf. aus mehreren Vokabularen zusammensetzt:
    - Mapping der Vokabulare untereinander (Artikulation) bzw. Harmonisierung
    - Abbildung der Daten auf die Vokabulare und ggf. Nutzung der zugehörigen Schemata
    - Nutzung von Zusatzwissen zu Orts- und Zeitbezug, auch als generische Zusammenhänge zwischen Daten
    - Harmonisierung der Darstellung von Daten aus verschiedenen Informationssystemen
    - Nutzung der in den Vokabularen enthaltenen Strukturen, z.B. zur Weiternavigation, Gruppierung, Facettierung etc.
- Beschreibung und Realisierung des technischen Zugriffs auf heterogene Informationssysteme (Datentypen, Schnittstellen und Formate)
- Nutzung generischer Komponenten zur Präsentation von Daten innerhalb eines Webportals
- Präsentation/Darstellung der Suchergebnisse in einer integrierten Trefferansicht
    - Koordination zwischen bzw. Orchestrierung von Komponenten
    - Möglichkeit zur Kommunikation zwischen Komponenten.

In Kapitel 2 wurde ausgehend von der gegebenen Zielstellung eine Grundarchitektur entwickelt, welche den Aufbau einer semantischen Suchfunktion auf einer allgemeinen, von technischen Randbedingungen unabhängigen Basis beschreibt.

Kapitel 3 beschreibt vier Varianten der Grundarchitektur. Ein erster Ansatz erweitert eine bestehende konventionelle Volltextsuche um Fachvokabular in Form eines Umweltthesaurus sowie die Möglichkeit zum Anschluss mehrerer Zielsysteme.
Die zweite Architekturvariante setzt auf die Vorverarbeitung der Suchanfrage sowie die Beschreibung und den Anschluss von Zielsystemen mittels Zielsystembeschreibungen über einen SearchBroker. Vokabulare werden durch ein Ontologiesystem verwaltet und über eine Artikulationsontologie miteinander verknüpft. Die Generierung der Ergebnis-

präsentation findet in einer Mashup-Komponente statt. Alle Komponenten werden dabei serverseitig betrieben.
Die dritte Variante realisiert eine serviceorientierte Architektur, verlagert wesentliche Teile der Anwendung auf das Client-System, während die Bereitstellung von Daten durch einen Webcache erfolgt, der Daten redundant vorhält und mit Hilfe generischer Dienste über einheitliche APIs bereitstellt. Die Harmonisierung der Darstellung und der Semantik der Daten aus den Zielsystemen findet in einem Transformationsschritt zwischen Zielsystem und Webcache statt. Die vierte Variante erweitert die vorhergehende um Dienste zur Verknüpfung von Daten aus verschiedenen Zielsystemen und bietet damit die Voraussetzungen für eine Datenbereitstellung und -nutzung im Sinne des Semantic Web.

In Kapitel 4 werden konkrete Umsetzungsbeispiele der Architekturvarianten präsentiert, bei denen es sich teilweise um Systeme im produktiven Einsatz handelt. Der Energieatlas Baden-Württemberg nutzt im Wesentlichen generische Frontendkomponenten, die ereignisbasierte Kommunikation zwischen den Komponenten mit Hilfe des zugehörigen Eventbus sowie Teile des Webcache. Die Landesumweltportale nutzen zusätzliche Mechanismen zur semantischen Erweiterung der Suchanfragen und setzen darüber hinaus auf die Verknüpfung von Daten bezüglich ihres Geobezugs. Mit Geoinformationen, Messwerten, Volltextsuche, Assets, Objektinformationen und Metadaten kommt hier die komplette Palette generischer Datentypen und zugehöriger Anzeigekomponenten zum Einsatz. Darüber hinaus wird die Nutzung von Zielsystembeschreibungen zur Implementierung einer flexiblen, dynamischen mobilen Anwendung anhand der App „Meine Umwelt“ dargestellt.

Die Gegenüberstellung und Diskussion der verschiedenen Architekturvarianten findet in Kapitel 5 statt.

Die wesentlichen Ergebnisse der Arbeit sind:

1. Entwicklung eines neuen Konzepts zur semantischen Suche in heterogenen Informationssystemen zu Fragestellungen aus den Bereichen „Umwelt“ und „Energie“.
2. Herleitung einer allgemeinen (generischen) Architektur, aus der sich vier verschiedene Architekturvarianten ableiten lassen.
3. Ableitung der ersten Architekturvariante durch Erweiterung der bestehenden Volltextsuche um domänenspezifisches Fachvokabular sowie um Möglichkeiten zum Zugriff auf weitere Zielsysteme auf Basis der OneBox-Schnittstelle.
4. Erarbeitung der zweiten Architekturvariante als serverseitigen Ansatz, in dem ein SearchBroker eine Vorverarbeitung der Suchanfrage vornimmt, das gesamte Hintergrundwissen in Form mehrerer Teilontologien abgelegt und mithilfe einer Artikulationsontologie miteinander vernetzt. Die Ergebnispräsentation wird Schablonen-basiert ebenfalls serverseitig erzeugt.
5. Ableitung einer dritten Architekturvariante, die Daten aus den Zielsystemen redundant über eine serviceorientierte Architektur („Webcache“) bereitstellt. Ein Transformationsprozess (Data Ingestion) sorgt für die Aufbereitung und seman-

tische Annotation der Daten. Die Suche und die Präsentation der Suchergebnisse sind über eine Sammlung von clientseitigen Komponenten realisiert, welche die notwendigen Anfragen asynchron an den Webcache richten. Die Gesamtanwendung wird mittels eines Kommunikationsbusses (Event-Bus) orchestriert.

6. Ableitung der vierten Architekturvariante durch Erweiterung der Funktionalität der Webcache-Services um semantische Technologien und Datenformate zur Realisierung von Linked Data als Grundlage des Semantic Web. Hinzufügen von Funktionalität zur Generierung neuer Inhalte in Form von Verknüpfungen zwischen Datenobjekten.
7. Implementierungen der ersten drei Architekturvarianten anhand von fünf konkreten Systemen im Umweltinformationssystem Baden-Württemberg.
8. Experimentelle Erprobung anhand von 18 definierten Szenarien aus dem Umwelt- und Energiebereich.
9. Ableitung von Aussagen über die Leistungsfähigkeit des vorgestellten Konzepts.
10. Bereitstellung einer leistungsfähigen semantischen Suche auf Basis der dritten Architekturvariante im Bereich des Umweltinformationssystems Baden-Württemberg als Beispiel für die Suche in heterogenen Informationssystemen innerhalb der Domänen „Umwelt“ und „Energie“.
11. Produktive Systeme, u.a. Landesumweltportale in Baden-Württemberg und vier weiteren Bundesländern, Energieatlas Baden-Württemberg und mobile App „Meine Umwelt“, basieren auf den entwickelten Architekturvarianten, insbesondere der dritten.

Das vorliegende Konzept bietet Spielraum für künftige Erweiterungen, zunächst insbesondere für die vollständige Umsetzung der Erweiterungen aus der vierten Architekturvariante.

Darüber hinaus gibt es jedoch Potenzial für weitere Verbesserungen:

Wenn Zielsysteme mit einer großen Zahl unterschiedlicher Vokabulare bzw. unterschiedlicher semantischer Darstellungen der Daten verwendet werden, sollte eine technische Unterstützung beim Zusammenführen bzw. beim Mapping der Vokabulare stattfinden, um den Prozess vollständig oder teilweise automatisieren zu können. Hier gibt es Ansätze (Ehrig et al. 2005; Noy 2009; Euzenat und Shvaiko 2007) zum (halb-) automatischen Erkennen der Semantik von Informationssystemen, z.B. anhand formaler Schemata.

Mit der eindeutigen Adressierbarkeit der Daten über eindeutige RESTful URLs ist eine wesentliche Voraussetzung für die Verwendung einer großen Menge von Technologien des Semantic Web gegeben. Um eine globale Verknüpfbarkeit und Nutzbarkeit der Daten zu gewährleisten, ist es jedoch notwendig, die Daten auf global gültige Schemata abzubilden. Auch wenn es Kritik bezüglich der Nutzung von Ontologien und XML gibt (Shirky 2005; Swartz 2013), so gibt es leichtgewichtige bzw. pragmatische Ansätze (Mikroformate, Microdata), die jedoch ebenso die gegenseitige Nutzung von Daten und

damit die Interoperabilität von Anwendungen zum Ziel haben und bereits von kommerziell erfolgreichen Suchmaschinen unterstützt/genutzt werden.

Aus der entwickelten Architektur lassen sich nicht zuletzt Empfehlungen und Vorschläge für das Aufsetzen künftiger Informationssysteme ableiten, um eine möglichst einfache Integrierbarkeit von deren Inhalte in übergreifende Informationssysteme (Webportale) zu gewährleisten sowie eine Interoperabilität mit dritten Anwendungen bieten zu können.

Anhang: Grundlagen

A1 Das Semantic Web

Eine Vision des Semantic Web (Berners-Lee und Fischetti 1999; Berners-Lee et al. 2001) ist die Bereitstellung von Informationen in einer Art und Weise, die es ermöglicht, Informationen maschinell zu verarbeiten, zu kombinieren und Fragen durch entsprechende Schlussfolgerungsmechanismen zu beantworten bzw. mit deren Hilfe sogar weitere Informationen zu generieren.

Hierzu gibt es bereits eine ganze Reihe etablierter Technologien. Ihre gemeinsame Grundlage sind

- Linked Data,
- Vokabulare,
- Abfragen (Queries) und
- Inferenzen (Schlussfolgerungen).

A1.1 Linked Data

Berners-Lee schlug in seinen Überlegungen zu einem Semantischen Web (Semantic Web) (Berners-Lee und Fischetti 1999) das Konzept von Linked Data vor (Berners-Lee 2006). Danach sollen alle Objekte über eindeutige Bezeichner (URIs) repräsentiert werden. Solche eindeutig bezeichneten Objekte werden Ressourcen genannt. Für das Semantic Web wird die Verwendung von HTTP-URIs als Bezeichnerformat vorgeschlagen, so dass sich Informationen zu einem Objekt auch tatsächlich nachschlagen lassen, d.h. die HTTP-URI adressiert einen Server, der Informationen zu einem Objekt in einem standardisierten Format liefert. Die Informationen enthalten im Regelfall weitere URIs, die beispielsweise Beziehungen des Objekts zu anderen Ressourcen, d.h. weiteren Objekten, die mit den gelieferten URIs bezeichnet sind, ausdrücken.

Beziehungen zwischen Objekten werden im Semantischen Web als Tripel dargestellt. Darin werden ein Subjekt und ein Objekt bezüglich eines Prädikates miteinander verknüpft.

Versteht man die Objekte als Knoten und die Prädikate als Kanten eines Graphen, so spannen die Beziehungen zwischen allen möglichen Objekten einen gigantischen Graphen auf, der im Semantischen Web auch als „Giant Global Graph“ (Berners-Lee 2007) bezeichnet wird.

Zur Definition von Ressourcen und für die Darstellung von Beziehungen zwischen Objekten gibt es eine Reihe von standardisierten Formaten, insbesondere das Ressource Description Format (RDF) sowie Varianten davon (RDFa, RDF/XML), die z.B. eine Einbettung von semantischen Informationen als Linked Data innerhalb von für den

menschlichen Nutzer bestimmten HTML-Dokumenten ermöglichen. Das Semantische Web muss also nicht notwendigerweise parallel zum herkömmlichen Web entstehen, sondern kann sich damit gewissermaßen überschneiden.

A1.2 Vokabulare

Um Daten und Objekte und den Beziehungen unter ihnen einen bestimmten Sinn zuordnen zu können, ist es notwendig, eine Konzeptionalisierung gleichartiger Objekte vorzunehmen und festzulegen, welche Beziehungen Objekte bzw. Konzepte untereinander haben können (Harras et al. 1991).

Eine solche Definition wird Vokabular genannt. Ein Vokabular betrifft in der Regel einen speziellen und abgegrenzten Gegenstandsbereich (Domäne). Bereits innerhalb einer kleinen Anwendung werden üblicherweise mehrere Vokabulare benötigt. Vokabulare werden nicht zentral verwaltet, müssen daher selbstbeschreibend sein. Die technische Umsetzung von Vokabularen für das Semantic Web erfolgt üblicherweise durch eine Definition in RDF Schema (RDF(S)) (Brickley und Guha 2014) oder in der Web Ontology Language (OWL) (W3C OWL Working Group 2012).

Vokabulare von übergreifender Bedeutung können standardisiert sein und wiederverwendet werden. Initiativen wie schema.org (schema.org 2016b) bieten bereits Vokabulare mit einer großen Auswahl nutzbarer und erweiterbarer Konzepte, die eine einheitliche Nutzung von Daten über Systemgrenzen hinweg ermöglichen sollen. Eigene ggf. anwendungsspezifische Vokabulare können vollständig neu definiert oder mit Hilfe von bestehenden Vokabularen wie SKOS (Simple Knowledge Organization System) (W3C 2009b, 2009c, 2009c) aufgebaut werden.

Im Semantic Web werden Vokabulare in Form von Ontologien beschrieben, die eine explizite Spezifikation einer Konzeptionalisierung darstellen.

A1.3 Abfragen (Queries)

Um die auf Basis von Vokabularen und Linked Data-Technologien semantisch beschriebenen Daten tatsächlich nutzen zu können, müssen sie gespeichert und abgefragt werden können. Das World Wide Web Consortium (W3C) hat hierfür eine Abfragesprache standardisiert, die „SPARQL Query Language“ (W3C 2008). SPARQL-Abfragen werden ähnlich RDF in Tripel übersetzt, die allerdings statt Werten auch Variablen enthalten können und deshalb Tripel-Muster genannt werden. Eine SPARQL-Maschine hat die Aufgabe, alle zu solchen Mustern passenden Ressourcen zurückzuliefern. Die meisten SPARQL-Maschinen speichern Daten mit Hilfe semantischer Datenbanken (Triplestores). SPARQL-Abfragen können z.B. über das HTTP-Protokoll oder per Webservice-Anfragen via SOAP (Simple Object Asses Protocol) (Mitra und Lafon 2007) transportiert werden.

A1.4 Inferenzen (Schlussfolgerung)

Wenn Daten auf Basis von Vokabularen und Linked Data dargestellt werden, können aus ihnen und ihren Beziehungen automatisiert neue Beziehungen gewonnen werden (Semantic Reasoning). Dazu benötigt man Zusatzinformationen, z.B. erweiterte Vokabulare (Ontologie-Sprachen) oder spezielle Regeln (Beschreibungslogik). Die meisten Inferenzmaschinen (Semantic Reasoner) nutzen die Prädikatenlogik erster Stufe (Sowa 2014).

Inferenzen können also dazu dienen, weitere, in den ursprünglichen Daten nicht enthaltene Beziehungen zu erzeugen. Die Zusatzinformationen bzw. Regeln können jedoch auch helfen, Inkonsistenzen in den Daten zu finden und zu eliminieren.

A2 Datentypen und der Strukturierungsgrad von Daten

Die Vielfalt der Datentypen lässt sich nach dem Grad ihrer Strukturierung unterscheiden. Für die maschinelle Verarbeitung von Daten, aber auch für das Verständnis der Daten durch den Menschen, ist eine systematische Strukturierung der Daten nicht nur hilfreich, sondern teilweise zwingend notwendig. Viele Daten liegen daher in strukturierter Form vor, z.B. in relationalen Datenbanksystemen (Lang und Lockemann 1995; Ritchie 2002; Sikos 2015), in denen Datensätze in Tabellen abgelegt werden, deren benannte und typisierte Spalten die Struktur eines Datensatzes und damit des semantischen Objektes ergeben. Ein Datenbankschema beschreibt also die Semantik der in der Datenbank enthaltenen, strukturierten Objekte. Häufig liegen erreichbare Daten jedoch nicht in strukturierter Form vor, sondern lediglich in einer schwachen oder völlig unstrukturierten Repräsentation. Ein Beispiel für typischerweise schwach strukturierte Daten stellen klassische HTML-Dokumente dar, in denen die inhaltliche Auszeichnungen in Form von sog. Markup-Tags erfolgt, z.B.

```
<title>Schnelllaufzahl eines Dreiblattrotors</title>
```

für den Titel eines Dokuments. Schwach strukturiert bedeutet, dass die Auszeichnung von Inhalten einer typischen Dokumentstruktur (Titel, Überschriften, Absätze) entspricht, nicht jedoch spezifischen Attributen aus der semantischen Domäne des Dokuments[22].

In vielen Dokumenten fehlt sogar die schwache Auszeichnung von Inhalten, man spricht dann von unstrukturierten Dokumenten bzw. unstrukturierten Daten. Das lässt sich in HTML-Dokumenten beispielsweise bei der verbreiteten exzessiven Verwendung von `<DIV>` oder `<SPAN>`-Tags beobachten, die zwar eine hierarchische Strukturierung von Dokumenten erlauben, denen jedoch keine Bedeutung zugeordnet ist.

[22] Es gibt Mechanismen mit deren Hilfe auch Markup-Dokumenten (XML) eine Struktur und/oder Semantik zugeordnet werden kann, z.B. Dokumenttyp-Definitionen (DTD), XML-Schema oder über Microtagging-Mechanismen (schema.org 2016b, 2016c)

In vielen Fällen werden (stark) strukturierte Daten in weniger strukturierte Repräsentationen überführt, z.B. wenn eine (strukturierte) Datenbank als Basis für die Erzeugung von Webseiten (HTML-Dokumente) dient. Suchmaschinen können häufig nur auf die schwächer strukturierte Repräsentation zugreifen. Da eine Rücktransformation in die strukturierte Form im Allgemeinen nicht möglich ist (Hänsch 2014) steht den Suchmaschinen daher sehr häufig nicht die volle Semantik von Daten zur Verfügung.

Der Grad der Strukturierung von Daten hängt einerseits von der Ausprägung ihrer inhärenten Struktur, zur technischen Nutzung jedoch ebenfalls vom Grad der Standardisierung ihrer Strukturen, ab (Abbildung 29).

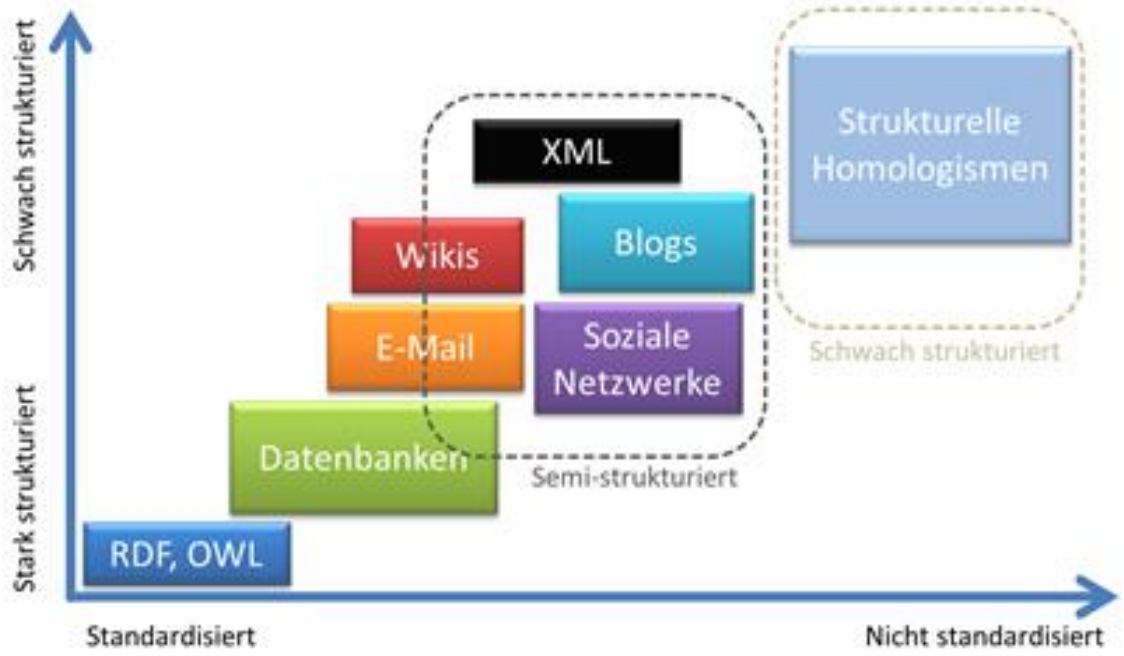

Abbildung 29: Strukturierung versus Standardisierung der Datenschemata; nach (Holzinger 2014)

Große Teile der im Internet verfügbaren Inhalte sind schwach strukturiert. Sie liegen zwar in standardisierten Datenformaten (meist HTML oder PDF) vor (Glöggler 2003), welche jedoch meist nicht die vollständige Semantik und Strukturierung der Informationen zum Ausdruck bringen, sondern sich häufig auf die allgemeine Struktur von Dokumenten (Titel, Überschriften, Absätze) beziehen, nicht jedoch auf den Inhalt. Teilweise werden schwach strukturierte Dokumente durch Metadaten ergänzt, die sie zu semi-strukturierten Daten machen können.
Andere Typen von Informationen sind stärker strukturiert, z.B. wenn in sozialen Netzwerken oder in Blog-Software bereits eine (grobe) Struktur für Inhalte vorgeben ist, und Autoren Informationen formularbasiert erstellen (semi-strukturierte Daten mit de-facto standardisierten Schemas, z.B. in sozialen Netzwerken). Auch die Nutzung bestimmter Protokolle (z.B. für Email) oder Datenformate (XML-Dokumente inklusive damit verbundenem XML-Schema) kann für eine bessere Strukturierung der Daten sorgen.

In den meisten verbreiteten Datenbanksystemen, insbesondere bei relationalen Datenbanken, muss im Datenbanksystem die Struktur der Daten durch Spezifikation von Schemas festgelegt werden. Die enthaltenen Daten sind hierdurch per se strukturiert. Die verwendeten Schemata können jedoch mehr oder weniger standardisiert sein, was wiederum den Datenaustausch mit anderen Systemen beeinflusst.
Sowohl stark strukturiert als auch standardisiert sind die Formate des Semantic Web wie RDF oder OWL.

A2.1 Grundlagen für die maschinelle Verarbeitung von Daten

Um die maschinelle (Weiter-)Verarbeitung der im Internet (WWW) vorhandenen Informationen zu ermöglichen, ist es notwendig, dass Maschinen den vorhandenen, heute häufig noch von Menschen zusammengetragenen Informationen deren Bedeutung (Semantik) eindeutig zuordnen können. Im Hinblick auf ein Internet der Dinge (Internet of Things) (Uckelmann et al. 2011b; Fleisch und Mattern 2005; Weber und Weber 2010) und den Einsatz von Computern in allen Bereichen des Lebens (Ubiquitous computing = „Rechnerallgegenwart“) (Greenfield 2006; Fleisch und Mattern 2005) ist es notwendig, menschliche Eingriffe (Interaktion) zu reduzieren oder sogar gänzlich überflüssig zu machen (Uckelmann et al. 2011a), d.h. eine Automatisierung durchzuführen.

Heute stehen Informationen in einer Vielzahl verschiedener elektronischer Systeme zur Verfügung. Informationssysteme stehen dabei in einem Beziehungsgefüge, das häufig als MAT-System (Mensch/Aufgabe/Technik-System), einem Dreieck zwischen Mensch, Aufgabe und (Informations-)Technik, bezeichnet wird. Die Ecken des Dreiecks betonen dabei verschiedene Schwerpunkte, die bei der Erstellung eines Informationssystems gesetzt werden können. Eine Verschiebung des Schwerpunktes in Richtung einer Ecke hat Einfluss auf die Betonung bzw. die Anforderungen an die jeweils anderen, z.B. bedeuten hohe Anforderungen im Bereich der Benutzbarkeit eines Informationssystem durch den Endnutzer meist auch erhöhte Anforderungen bei der Definition und Ausgestaltung der Aufgabe (funktionale Anforderungen) sowie bei der Auswahl der verwendeten Technologie.

Umgekehrt wird bei Informationssystemen, die sich am menschlichen Nutzer orientieren, häufig kein oder wenig Wert auf die maschinelle Interpretierbarkeit und Weiterverarbeitbarkeit von Informationen gelegt, da die notwendige Interpretation als (i.A. leistungsfähiger) kognitiver Prozess beim Nutzer abläuft. Wieder umgekehrt, stellt eine einfache maschinelle Verarbeitbarkeit von Daten hohe Anforderungen an die Daten bzw. deren Strukturierung.

Der Grad der Strukturierung von Daten in Informationssystemen ist in der Realität häufig vom Anwendungsfall und der Zielgruppe (der Nutzer) abhängig. Gerade Fachinformationen sind oft für den Konsum (die Verarbeitung) durch menschliche Nutzer aufbereitet, und die Informationssysteme entsprechend dafür implementiert und optimiert.

Viele Daten sind schwach strukturiert und liegen z.B. in Form von Dokumenten (PDF, DOC) und zugehörigen Metainformationen vor.

A2.2 Semantische Interpretation von Daten

Die unmittelbare semantische Interpretation von schwach strukturierten Daten ist häufig nur auf Basis der verfügbaren Metadaten (Titel, Inhaltsangabe, Stand, Gültigkeitsbereich, Schlagworte etc.) möglich, die zur Verwendung durch menschliche Nutzer auch zum Sortieren, Filtern und Gruppieren herangezogen werden können, jedoch, insbesondere wenn fachliche Metadaten vorhanden sind, ein inhaltliches Grundverständnis des Nutzers voraussetzen.

In der Praxis werden Dokumentenbestände als Paradebeispiel für große Mengen unstrukturierter Daten (unter weitgehender Umgehung ihrer Semantik) häufig durch Volltextsuchmaschinen erschlossen. Die meisten Volltextsuchen basieren dabei auf zuvor gebildeten Indexen und dem lexikalischen Vergleich von im Index gespeicherten Begriffen und den verwendeten Suchbegriffen.

Intelligentere Verfahren wie „Text Mining“ (Heyer et al. 2008; Witte und Mülle 2006; Khadjeh Nassirtoussi et al. 2014; Lemke et al. 2016) können inhaltliche Strukturen aus Dokumenten erkennen und dem Nutzer zusammengehörige Informationsblöcke liefern, von denen er zuvor nicht weiß, ob und wo sie in einem Dokumentenbestand enthalten sind.

Andere Typen von Daten können stark strukturiert sein, z.B. die Daten dauerhafter Messprogramme (Zeitreihen zur Luftqualität, Pegelstände oder Wetterdaten), die meist in festen Strukturen, z.B. mit Hilfe relationaler Datenbankmodelle, abgelegt werden. Die Struktur der Daten bleibt in der Regel auch beim Zugriff erhalten, z.B. werden einzelne Messwerte, der zugehörige Zeitstempel und Stammdaten (Ort der Messung, Messmethodik) jeweils zusammenhängend ausgeliefert.

Die semantische Interpretation ist im Allgemeinen jedoch auch bei stark strukturierten Daten abhängig von Zusatzinformationen, d.h. die Semantik ergibt sich nicht alleine aus den Daten selbst, sondern z.B. nur unter Einbeziehung von Metainformationen, die Aufschluss über die Bedeutung von Spaltenbezeichnern, verwendete Einheiten und Ähnliches geben.

A3 Webportale

Bei Webportalen handelt es sich um eine spezielle Art von Websites, bei denen die Bündelung von Informationen zu einem Themengebiet im Vordergrund steht, zum Beispiel um den Nutzern einen zentralen Einstieg in das Thema zu bieten. Typischerweise stammen die im Portal präsentierten Informationen aus unterschiedlichen Datenquellen, zum Beispiel in einem Flugbuchungsportal aus den Systemen verschiedener Fluggesellschaften. Daher ist es eine Aufgabe des Portals bzw. seiner Entwickler, Informationen zu bündeln, ggf. zu harmonisieren und dem Benutzer in geeigneter Form, meist

einheitlich, zu präsentieren, z.B. durch Selektionsmechanismen wie die Facettierung (Stock und Stock 2008). Häufig sind Webportale mit einer ganzen Reihe von Funktionalitäten ausgestattet, z.B. der Möglichkeit zur Personalisierung (Einstellungen) durch den Nutzer, interaktiven Elementen wie Foren oder Bewertungssystemen, Suchmaschinen etc.

A4 Serviceorientierte Architekturen

Serviceorientierte Architekturen (SOA) stellen eine moderne Form verteilter Informationssysteme dar und wurden erstmals 1996 beschrieben (OASIS Open 2006; Schulte und Natis 1996). Die Grundlage für SOAs bilden Beschreibungen von Geschäftsprozessen auf verschiedenen Abstraktionsebenen, die jeweils entsprechende, aufeinander aufbauende Implementierungen haben. Ein „höherwertiger“ Prozess besteht also in der Regel aus einer Zusammensetzung einfacherer Dienste/Prozesse, die er zielführend verwendet.

Ein wesentliches Prinzip der Architektur ist dabei die Wiederverwendbarkeit von Diensten, bei denen es sich jedoch immer um Teile des jeweiligen Geschäftsprozesses, also inhaltlichen Aufgaben, handelt – im Gegensatz zu rein technischen Aufgaben (wie das Ausführen einer einzelnen Datenbankabfrage).

Sichtbar und nutzbar werden die einzelnen Geschäftsprozesse (Dienste) durch ihre Schnittstellen (Eingaben/Parameter und Ergebnisse). Serviceorientierte Architekturen spielen eine große Rolle bei der Implementierung von Internet-Anwendungen, die dabei auf eine ganze Reihe standardisierter (technischer) Protokolle aufsetzen können.

Aus technischer Sicht hat sich in den letzten Jahren eine Veränderung der Technologien zur Implementierung der Serviceschnittstellen ergeben. Prägten noch vor wenigen Jahren klassische Webservices (Booth et al. 2004; Bettag 2001) und XML-Formate (Ray 2002; Harold 2002) die Dienste-Landschaft, kommen heute zunehmend leichtgewichtige Schnittstellen und Datenformate zum Einsatz, die auf das Programmierparadigma REST (Representational State Transfer) (Richardson 2007; Fielding 2000) und das JSON-Format (JavaScript Object Notation) (ECMA International 2013) setzen, welche beide direkt von Webbrowsern sowie serverseitigen Infrastrukturen unterstützt werden. Insofern stellen RESTful Services einen pragmatischen Ansatz der Maschine-zu-Maschine-Kommunikation dar, unter die auch die Kommunikation zwischen Frontends und Hintergrunddiensten fällt.

Eine verbreitete Definition von SOA (OASIS Open 2006) sieht vor, dass die einzelnen Dienste („verteilte Funktionalität“) einer SOA von mehreren Anbietern bereitgestellt werden können –also (zumindest) organisatorisch nur lose gekoppelt sind. Das erfordert eine bestimmte Generizität solcher Dienste, die sich durch Parametrisierbarkeit und die Standardisierung von Datenmodellen ausdrückt. Große Internet-Dienstleister wie Google bieten eine große Zahl solcher generischer Dienste an, z.B. Gazetteer-

Services zum Auflösen von Ortsnamen oder Standortdaten[23]. Im Sinne einer SOA erfüllen solche Dienste durchaus inhaltliche Aufgaben. Deren Nutzung in eigenen Anwendungen ist also gerade in Bezug auf die Wiederverwendbarkeit von Softwarekomponenten und damit die Vermeidung redundanter Implementierungen im Sinne des Entwickelns von verteilten Anwendungen nach der SOA-Architektur sinnvoll. Die rechtlichen, sicherheitstechnischen bzw. lizenztechnischen Aspekte solcher Lösungen sind relevant, jedoch nicht Bestandteil der vorliegenden Diskussion.

Häufig werden allgemein nutzbare Dienste über große Cloud-Infrastrukturen (Baun et al. 2010) (Weinhardt et al. 2009) zur Verfügung gestellt, was meistens ein hohes Maß an Verfügbarkeit und Skalierbarkeit sicherstellt[24].

Auch im Zusammenhang mit per se (oder de facto) verteilten Informationen bietet sich die Nutzung von serviceorientierten Infrastrukturen an. Daten und Prozesse, die noch nicht durch Dienste bereitgestellt werden, können häufig mit relativ geringem Aufwand als Adapter (Gamma 2004) (auch: „Wrapper“) an eine bestehende (nicht serviceorientierte) Anwendung gekoppelt werden. Wo eine solche Anpassung nicht möglich ist, reicht häufig auch die Entwicklung einer parallelen, ggf. abgespeckten, Variante der Original-Anwendung und einem Mechanismus zur Synchronisierung von Daten.[25]

Zur Implementierung serviceorientierter Architekturen hat sich inzwischen eine Reihe von Basistechnologien (technischen Protokollen) durchgesetzt. Allen gemeinsam ist dabei der grundsätzliche Anspruch der Plattformunabhängigkeit, die sich durch mehr oder weniger starke Standardisierung der Protokolle ausdrückt. Stärker standardisiert, dadurch jedoch auch schwergewichtiger, sind dabei Webservices (häufig auch: SOAP-Webservices), die in ihrem Technologie-Portfolio auch mit Dienstbeschreibungen (Web Services Description Language, WSDL) (W3C 2001, 2007) und Dienstverzeichnissen (Universal Description, Discovery and Integration, UDDI) (OASIS UDDI Specifications TC 2016) aufwarten können. In der Praxis kommen jedoch seit einigen Jahren immer häufiger die weniger standardisierten, dafür leichtgewichtigeren RESTful Services zum Einsatz, die wie auch die SOAP-Webservices im Wesentlichen auf Standard-Webtechnologien aufbauen, dabei jedoch weitgehend auf mit Standardisierung verbundenen Overhead verzichten (Bayer 2002).

Proprietäre, d.h. nicht standardisierte, Schnittstellen, stellen ein Hindernis beim Aufsetzen einer serviceorientierten Architektur im Kontext bestehender Systeme dar. Auch wenn proprietäre Schnittstellen im Hinblick auf eine einzelne Anwendung durchaus sinnvoll sein können, z.B. was ihre Effizienz betrifft, so hinderlich ist die fehlende In-

---

23 https://developers.google.com/maps/documentation/geocoding/

24 Das ist normalerweise Bestandteil eines (ggf. auszuhandelnden) Service-Level-Agreements (SLA) zwischen dem Anbieter und den Anwendern solcher Dienste.

25 Ein solches Vorgehen ist in der betrieblichen Praxis aufgrund von bestehenden Altsystemen häufig notwendig. Ziel sollte dabei jedoch immer die Einbindung von Systeme in die SOA sein, z.B. beim nächsten Update oder der Ablösung einer Anwendung.

teroperabilität, wenn solche Systeme geöffnet und in einem größeren Kontext verfügbar gemacht werden sollen, z.B. wenn Daten aus einem Fachsystem im Zuge des E-Government verfügbar gemacht werden sollen.

Wenn interoperable Schnittstellen nicht direkt in den Originalsystemen implementiert werden können, kann die Interoperabilität durch das Vorschalten von Adapter-Programmen hergestellt werden, die auf der einen Seite die proprietäre Schnittstelle implementieren, und alle oder einzelne Dienste auf der anderen Seite z.B. als Web- oder RESTful Services zur Verfügung stellen. Um dabei auch eine semantische Interoperabilität zu erreichen, müssen die Adapter im Allgemeinen auch eine Abbildung des proprietären Datenmodells auf ein standardisiertes oder zumindest in der SOA bekanntes Datenmodell vornehmen.

A4.1 Microservices

Microservices stellen eine spezielle Art von Diensten dar (Fowler und Lewis 2015). Ihr wesentlichstes Merkmal ist, dass jeder Microservice für einen speziellen, kleinen und abgegrenzten Aufgabenbereich verantwortlich ist („Do one thing and do it well!“)[26], und seine Funktion(en) über eine (möglichst sprachunabhängige) Schnittstelle zur Verfügung stellt. Microservices können sich über ihre Schnittstellen auch gegenseitig nutzen – sind aber grundsätzlich voneinander entkoppelt. Auf Microservices beruhende Anwendungen sind somit per se modular aufgebaut und einzelne Microservices sind leicht austauschbar. Bei der Entwicklung und Wartung bieten Microservices aufgrund ihrer überschaubaren Funktionalität und Größe erhebliche Vorteile: Kurze Entwicklungszeiten und die Möglichkeit des Einsatzes von Automatisierungswerkzeugen (Continuous Integration und/oder Continuous Delivery) (Fowler 2006; Humble et al. 2006).

[26] Das Zitat stammt ursprünglich von Douglas McIlroy, dem Erfinder der Unix-Pipes, und bezieht sich auf Unix-Kommandos. Es wird jedoch auch häufig im Zusammenhang mit Microservices verwendet.

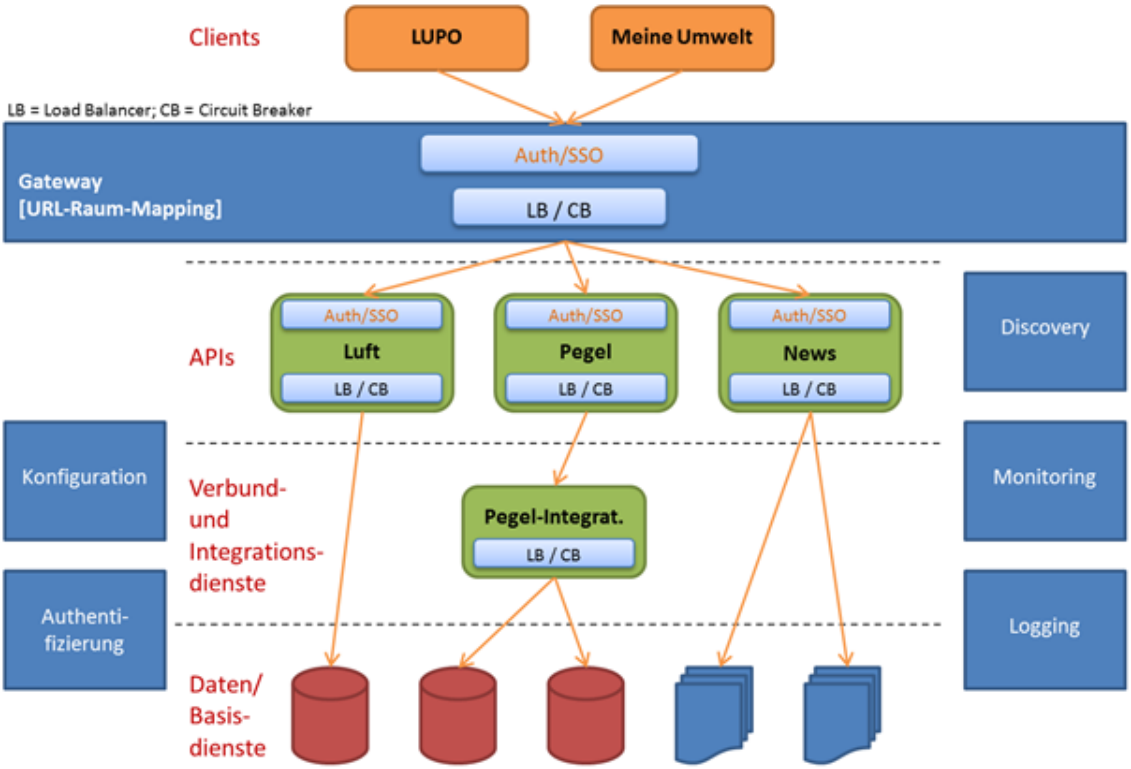

Abbildung 30: Beispiel einer Microservice-basierten Architektur für Landesumweltportale

Ein einfaches Beispiel einer Microservice-basierten Architektur ist in Abbildung 30 dargestellt. Über ein Gateway, das den zentralen Einstiegspunkt für nutzende Anwendungen darstellt, greifen Anwendungen (hier: LUPO, Meine Umwelt) auf eine Reihe von Microservices (Luft, Pegel, News) zu.

Unterhalb der eigentlichen Microservices stehen potentiell eine Reihe von Verbund- und Integrationsdiensten zur Verfügung, u.a. (in der Abbildung nicht dargestellt) die Möglichkeit zur ereignisgetriebenen Kommunikation der Microservices untereinander oder die Abstraktion beim Zugriff auf die Persistenzschicht (hier: Pegel-Integration), d.h. Daten- bzw. Basisdienste. Load Balancer (LB) und Circuit Breaker (CB) kümmern sich um die Verteilung der eingehenden Anfragen auf die einzelnen Services bzw. ihrer Instanzen. Zentrale Aufgaben, z.B. Monitoring, Logging, Konfigurationsmanagement, Authentifizierung und Autorisierung etc., werden durch eine Reihe von Infrastrukturdiensten übernommen, die der gesamten Plattform sowie allen Microservices zur Verfügung stehen.

A4.2 Schnittstellen und Protokolle

(SOAP-)Webservices

Bei den SOAP-Webservices handelt es sich um einen Standard des W3C-Konsortiums (W3C 2004d). Dabei stützt sich SOAP auf verschiedene Internet-Basistechnologien zum Transport von Daten, bietet darüber hinaus jedoch einen hohen Grad an Standar-

disierung, der sich vor allem in der Festlegung der verwendeten technischen Datenformate (meist XML-basiert) ausdrückt. Die semantische Interpretation beschränkt sich dabei auf die technische Verarbeitung von Daten. Was die eigentlichen (Nutz-)Inhalte betrifft, macht SOAP keine Vorgaben, d.h. die inhaltliche Festlegung und Interpretation von Daten erfolgt anwendungsspezifisch.

Da der Datenaustausch mittels SOAP de facto XML-basiert ist, entsteht durch das Aufbereiten/Einpacken, den Transport, das Auspacken und Validieren von Daten ein teilweise erheblicher Aufwand (Trapp 2007).

Allerdings bietet SOAP auch Unterstützung komplexer, aus mehreren Teilen bestehender Anfragen, so dass sich der Aufwand in manchen Anwendungsszenarien wieder relativieren kann.

Technisch gesehen kann der Austausch von SOAP-Nachrichten verschiedene Transport-Protokolle nutzen, neben dem verbreiteten HTTP/HTTPS-Protokoll des World Wide Web auch solche für den Dateitransfer (File Transfer Protocol, FTP) oder sogar Email-Protokolle (Simple Mail Transfer Protocol, SMTP). Die (normalerweise) XML-basierten Nachrichten umfassen ein System von Umschlägen („Envelopes“), in denen Metadaten („Header“) und Nutzdaten („Body“) enthalten sind (W3C 2004d).

RESTful Services (REST)

Representational State Transfer (REST) stellt keine Normierung oder Standardisierung im engeren Sinne dar. Es handelt sich vielmehr um ein Programmierparadigma, das als Konvention für die Verwendung des HTTP-Protokolls durch Programme verstanden werden kann (Bayer 2002).

Im Gegensatz zur Verwendung des WWW durch menschliche Nutzer dienen REST-Services dem Datenaustauch zwischen Programmen, d.h. die übermittelten Daten dienen der Weiterverarbeitung durch die empfangende Anwendung[27].

REST bedient sich dabei des HTTP-Protokolls und reduziert die möglichen Operationen zwischen Anwendungen auf eine kleine Menge von Operationen, die im Wesentlichen den CRUD-Operationen[28] für Datenbankanwendungen entsprechen, welche dabei im Wesentlichen den HTTP-Operationen PUT/POST, GET, PATCH/PUT bzw. DELETE zugeordnet werden.

Der Aufbau von REST-URLs ist nicht standardisiert, es gibt jedoch Best-Practise-Ansätze, die Empfehlungen für den über die URL-Syntax hinausgehenden Aufbau der REST-URLs geben[29] (Fredrich 2013).

---

[27] Die Anwendung kann allerdings auch im Webbrowser eines Anwenders laufen und z.B. eine Ansicht für einen menschlichen Nutzer generieren.

[28] C für das Anlegen (**C**reate), R für das (wiederholte) Lesen (**R**ead), U für das (ggf. wiederholte) Aktualisieren (**U**pdate) und D für das Löschen eines Datensatzes (**D**elete).

[29] http://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api#restful

Zum Beispiel liefert die GET-Anfrage auf die URL

```
http://services-bw.de/water/gauging
```

eine Liste aller Pegelstationen. Die URL

```
http://services-bw.de/water/gauging/177
```

liefert die Daten zu einer bestimmten Pegelstation mit der ID 177. Darüber hinaus können die Anfragen über weitere URL-Parameter genauer spezifiziert werden, z.B. schränkt die URL

```
http://services-bw.de/water/gauging/177?start=2013-05-
31&end=2013-06-03
```

die Anfrage auf einen bestimmten Zeitraum ein.

Auch die Datenformate für den Datenaustausch mittels RESTful Services sind nicht standardisiert. Es kommen häufig XML-basierte oder JSON-Formate zum Einsatz, jedoch auch HTML. Seit der Einführung von Ajax (Garrett 2005) und HTML5 (Hickson et al. 2014) wird zunehmend JSON (ECMA International 2013) verwendet, da das leichtgewichtige Format in Webbrowsern direkt durch JavaScript-Programme verarbeitet werden kann.

Im Allgemeinen können die durch einen RESTful Service gelieferten Daten Adressen (URLs bzw. URIs) für weitere REST-Aufrufe enthalten, die z.B. Beziehungen zwischen Objekten oder Kompositionen (ein Objekt besteht aus mehreren anderen Objekten) ausdrücken können. Eine solche Darstellung von Beziehungen wird auch als HATEOAS (**H**ypermedia **a**s **t**he **E**ngine **o**f **A**pplication **S**tate) bezeichnet und stellt ein wichtiges Prinzip von REST-basierten Anwendungen dar (Wikipedia 2016). HATEOAS ist auch einen Brückenschlag zu „Linked Data“ aus dem Bereich des Semantic Web.

A5 Cloud-Dienste

Viele Anwendungen und Dienste werden inzwischen nicht mehr auf dedizierten Servern zur Verfügung gestellt, sondern nutzen große Server-Infrastrukturen, die Ressourcen nach Bedarf dynamisch zuteilen. Das führt aus Sicht der Betreiber unter Anderem zu einer erhöhten Effizienz, Skalierbarkeit und Ausfallsicherheit.

Cloud-Dienste lassen sich grob in drei Klassen unterscheiden (Abbildung 31):

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS).

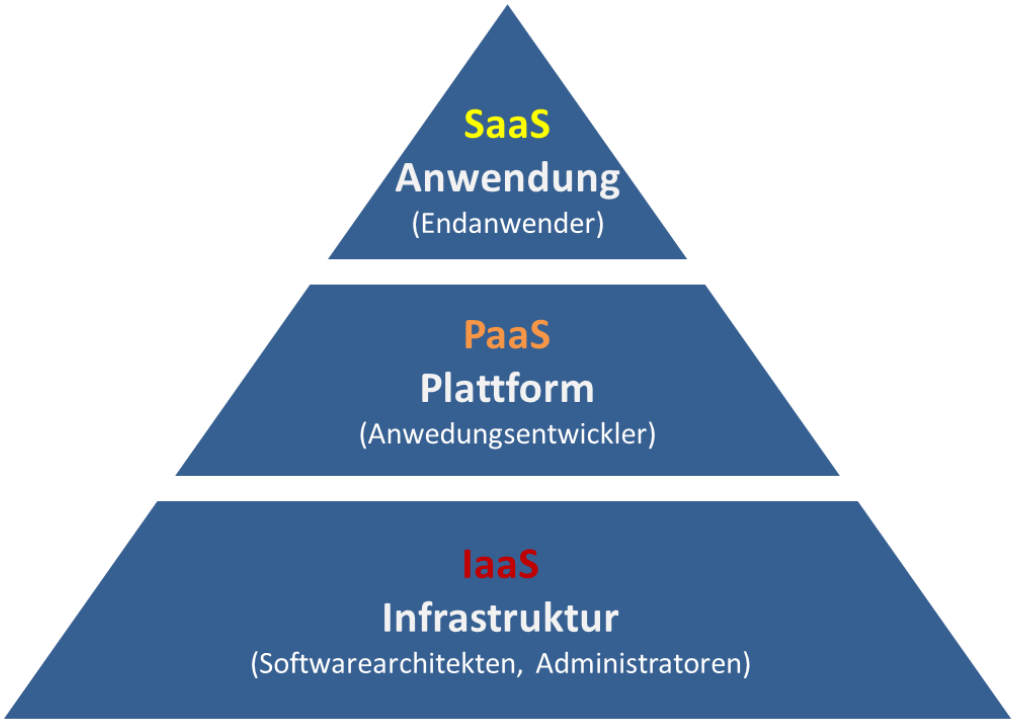

Abbildung 31: Cloud-Pyramide (nach http://skalicloud.com/v4/the-cloud-pyramid/)

Die Klasse „Infrastructure as a Service“ (IaaS) beschreibt Dienste eines Cloud-Anbieters, die Rechner-, Datenspeicherungs- und Netzwerkinfrastruktur bereitstellen. Beispiele hierfür sind die „Amazon Web Services“ oder die „Google Compute Engine“ (Leong et al. 2017).

Zu der Softwareschicht „Platform as a Service“ (PaaS) zählt man Angebote von Cloud-Anbietern, die die eigene Programmierung und Bereitstellung von Anwendungen ermöglichen und hierfür weitere Hilfsdienste bereitstellen. Bei Google ist das wichtigste PaaS-Angebot die „App Engine“, eine Programmier- und Laufzeitumgebung für Web- und serviceorientierte Anwendungen. Dort gehostete Programme können dabei Dienste aus der IaaS-Ebene nutzen. Auf der PaaS-Ebene stehen ebenfalls weitere Dienste, z.B. der „BigQuery“-Service zur Analyse großer Datenbestände zur Nutzung durch „App Engine“-Programme zur Verfügung.

Die Schicht „Software as a Service“ (SaaS) der Cloud-Pyramide steht schließlich für die Bereitstellung von Endanwendungen für Kunden in der Cloud. Bei Google umfassen die SaaS-Angebote z.B. die „Google Business Apps“, die eine Office-ähnliche Funktionalität bereitstellen, die über Webbrowser und mobile Anwendungen genutzt werden kann, sowie die Google-Maps-Lösungen, über die fertige Tools, Werkzeuge und APIs zur Bearbeitung und Darstellung von Geodaten angeboten werden. Viele der SaaS-Lösungen stellen ebenso wie die Speicherlösungen auf der IaaS-Ebene neben der reinen Endnutzerfunktionalität zusätzlich APIs bereit, die in eigenen Anwendungen auf der PaaS-Ebene oder für externe Anwendungen genutzt werden können (Schlachter et al. 2014a).

Eidesstattliche Versicherung

Die vorliegende Arbeit wurde von mir selbstständig angefertigt und es wurden keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Wörtlich oder inhaltlich übernommenen Stellen wurden als solche kenntlich gemacht. Die Satzung des Karlsruher Instituts für Technologie (KIT) zur Sicherung guter wissenschaftlicher Praxis in der gültigen Fassung wurde beachtet.

Ort, Datum

Unterschrift

Literaturverzeichnis

Abecker, Andreas; Bock, Jürgen; Kleb, Joachim; Wissmann, Jens; Bügel, Ulrich; Chaves, Fernando et al. (2009a): SUI - Ein Demonstrator zur semantischen Suche im Umweltportal Baden-Württemberg. In: Roland Mayer-Föll (Hg.): UIS Baden-Württemberg. F+E Vorhaben KEWA. Kooperative Entwicklung wirtschaftlicher Anwendungen für Umwelt, Verkehr und benachbarte Bereiche in neuen Verwaltungsstrukturen Phase IV 2008/09 : Wissenschaftliche Berichte. Karlsruhe: FZKA, S. 157–166.

Abecker, Andreas; Bock, Jürgen; Kleb, Joachim; Wissmann, Jens; Bügel, Ulrich; Chaves Fernando et al. (2009b): Ein Prototyp zur semantischen Suche im Umweltportal Baden-Württemberg. In: Roland Mayer-Föll (Hg.): UIS Baden-Württemberg. F+E Vorhaben KEWA. Kooperative Entwicklung wirtschaftlicher Anwendungen für Umwelt, Verkehr und benachbarte Bereiche in neuen Verwaltungsstrukturen Phase IV 2008/09 : Wissenschaftliche Berichte, Bd. 7500. Karlsruhe: FZKA, S. 157–166.

Abecker, Andreas; Bügel, Ulrich; Ebel, Renate; Schlachter, Thorsten (2011): Integrating semantic search for relational data into environmental information systems. In: Werner Pillmann (Hg.): Innovations in Sharing Environmental Observations and Information : EnviroInfo 2011 : 25th Internat.Conf.on Environmental Informatics, Ispra, I. Aachen: Shaker, S. 321–329.

Amazon (2017): Amazon Alexa. Online verfügbar unter https://developer.amazon.com/alexa, zuletzt geprüft am 02.04.2017.

Angrick, Michael; Bös, Richard; Rüther, Maria; Bandholtz, Thomas (2002): Semantic Network Services (SNS). In: Klaus Tochtermann Werner Pillmann (Hg.): IGU/ISEP: IGU/ISEP, S. 78–84.

Apache Cordova (2017): Apache Cordova. Online verfügbar unter https://cordova.apache.org/, zuletzt geprüft am 08.10.2017.

Apache Jena (2016): Apache Jena. The Apache Software Foundation. Online verfügbar unter https://jena.apache.org/, zuletzt geprüft am 06.10.2016.

Apple Inc. (2017): iOS, Siri. Online verfügbar unter https://www.apple.com/ios/siri/, zuletzt geprüft am 02.04.2017.

Baden-Württemberg, Landtag (2013): Gesetz zur Förderung des Klimaschutzes in Baden-Württemberg. Online verfügbar unter https://www.landtag-bw.de/files/live/sites/LTBW/files/dokumente/gesetzblaetter/2013/GBl201311.pdf, zuletzt aktualisiert am 30.07.2013, zuletzt geprüft am 04.10.2016.

Bandholtz, Thomas: Implementation of a Semantic Network Service (SNS) in the context of the German Environmental Information Network (gein®). In: SWDB'03 Proceedings of the First International Conference on Semantic Web and Database, S. 177–

189. Online verfügbar unter https://dl.acm.org/citation.cfm?id=2889917, zuletzt geprüft am 01.06.2018.

Baun, Christian; Kunze, Marcel; Nimis, Jens; Tai, Stefan (2010): Cloud Computing. Web-basierte dynamische IT-Services. Berlin: Springer (Informatik im Fokus). Online verfügbar unter http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10351852.

Bayer, Thomas (2002): REST Web Services - Einführung u. Vergleich mit SOAP. Orientation in Objects GmbH. Online verfügbar unter https://www.oio.de/public/xml/rest-webservices.htm, zuletzt geprüft am 02.06.2016.

Bergman, Michael K. (2001): White Paper. The Deep Web: Surfacing Hidden Value. In: *The Journal of Electronic Publishing* 7 (1). DOI: 10.3998/3336451.0007.104.

Berners-Lee, Tim (2006): Linked-Data. Online verfügbar unter https://www.w3.org/DesignIssues/LinkedData.html, zuletzt aktualisiert am 18.06.2009, zuletzt geprüft am 02.04.2017.

Berners-Lee, Tim (2007): Giant Global Graph. Online verfügbar unter http://dig.csail.mit.edu/breadcrumbs/node/215, zuletzt geprüft am 20.06.2016.

Berners-Lee, Tim; Fischetti, Mark (1999): Weaving the Web. The original design and ultimate destiny of the World Wide Web by its inventor. 1. ed. San Francisco, Calif.: HarperSanFrancisco. Online verfügbar unter http://www.loc.gov/catdir/description/hc044/99027665.html, zuletzt geprüft am 08.05.2017.

Berners-Lee, Tim; Hendler, James; Lassila, Ora (2001): The semantic web. In: *Scientific american* 284 (5), S. 28–37. Online verfügbar unter https://www-sop.inria.fr/acacia/cours/essi2006/Scientific%20American_%20Feature%20Article_%20The%20Semantic%20Web_%20May%202001.pdf, zuletzt geprüft am 02.04.2017.

Bettag, Urban (2001): Web-Services. Gesellschaft für Informatik (GI). Informatiklexikon. Online verfügbar unter https://www.gi.de/service/informatiklexikon/detailansicht/article/web-services.html, zuletzt geprüft am 20.06.2016.

Bizer, Christian; Heath, Tom; Berners-Lee, Tim (2009): Linked Data - The Story So Far. In: *International Journal on Semantic Web and Information Systems* 5 (3), S. 1–22. DOI: 10.4018/jswis.2009081901.

blak-it.com (2017): Single-page applications vs. multiple-page applications: pros, cons, pitfalls. Online verfügbar unter https://blak-it.com/blog/spa-advantages/, zuletzt geprüft am 08.10.2018.

Booth, David; Haas, Hugo; McCabe, Francis; Newcomer, Eric; Champion, Michael; Ferris, Chris; Orchard, David (2004): Web Services Architecture. W3C Working Group Note 11 February 2004. W3C. Online verfügbar unter https://www.w3.org/TR/ws-arch/, zuletzt geprüft am 20.06.2016.

Brickley, Dan; Guha, R. V. (2014): RDF Schema 1.1. W3C Recommendation 25 February 2014. W3C. Online verfügbar unter https://www.w3.org/TR/rdf-schema/, zuletzt geprüft am 20.06.2016.

Brutlag, Jake (2009): Speed Matters for Google Web Search. Google Inc. Online verfügbar unter http://services.google.com/fh/files/blogs/google_delayexp.pdf, zuletzt aktualisiert am 22.06.2009, zuletzt geprüft am 14.12.2017.

Bry, Francois; Kraus, Michael; Olteanu, Dan; Schaffert Sebastian (2001): Aktuelles Schlagwort "Semi-strukturierte Daten". Online verfügbar unter http://www.en.pms.ifi.lmu.de/publications/PMS-FB/PMS-FB-2001-9.pdf, zuletzt aktualisiert am 13.06.2001, zuletzt geprüft am 02.10.2017.

Bügel, U.; Schmieder, M.; Schnebel, B.; Schlachter, T.; Ebel, R. (2011a): Leveraging Ontologies for Environmental Information Systems. In: *Environmental Software Systems. Frameworks of eEnvironment*, S. 364–371.

Bügel, Ulrich; Chaves, Fernando; Düpmeier, Clemens; Schlachter, Thorsten; Weidemann, Rainer; Briesen, Marcus et al. (2010): SUI II - Weiterentwicklung der diensteorientierten Infrastruktur des Umweltinformationssystems Baden-Württemberg für die semantische Suche nach Umweltinformationen. In: Roland Mayer-Föll, Renate Ebel und Werner Geiger (Hg.): Umweltinformationssystem Baden-Württemberg. F+E-Vorhaben KEWA; Phase V, 2009/10. Karlsruhe, Baden: Universität Karlsruhe Universitätsbibliothek (KIT Scientific Reports, 7544), S. 43–50.

Bügel, Ulrich; Chaves, Fernando; Schmieder, Martin; Schnebel, Boris; Düpmeier, Clemens; Schlachter, Thorsten et al. (2011b): SUI für Umweltportale - Entwurf und prototypische Implementierung einer Architektur für die semantische Suche im Portal Umwelt-BW. In: Roland Mayer-Föll, Renate Ebel und Werner Geiger (Hg.): Umweltinformationssystem Baden-Württemberg - UIS BW. F+E-Vorhaben KEWA; Phase VI, 2010/11. Print on demand. Karlsruhe: KIT Scientific Publ (KIT Scientific Reports, 7586), S. 21–32. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/servlet/is/100252/kewa6-iosb-sui.pdf?command=downloadContent&filename=kewa6-iosb-sui.pdf.

Bügel, Ulrich; Schmieder, Martin; Schnebel, Boris; Schlachter, Thorsten; Ebel, Renate (2011c): Leveraging Ontologies for Environmental Information Systems. In: Jiří Hrebicek (Hg.): Environmental Software Systems : Framework of eEnvironment ; 9th IFIP WG 5.11 Internat.Symp. (ISESS 2011) : IFIP Advances in Information and Communication Technology; 359, Bd. 359: Springer, S. 364–371.

Bundesministerium für Wirtschaft und Energie (2017): THESEUS. Online verfügbar unter http://www.digitale-technologien.de/DT/Navigation/DE/Service/Abgelaufene_Programme/THESEUS/theseus.html, zuletzt geprüft am 08.10.2017.

Buxmann, Peter (1996): Standardisierung betrieblicher Informationssysteme (Gabler Edition Wissenschaft). Online verfügbar unter http://dx.doi.org/10.1007/978-3-663-08966-7.

CARTO (2017): Location Intelligence Software - CARTO. Online verfügbar unter https://carto.com/, zuletzt geprüft am 08.10.2017.

Christ, Oliver (2003): Content-Management in der Praxis. Erfolgreicher Aufbau und Betrieb unternehmensweiter Portale. Berlin, Heidelberg: Springer Berlin Heidelberg; Imprint; Springer (Business Engineering).

Conrad, Stefan (1997): Föderierte Datenbanksysteme. Konzepte der Datenintegration. Berlin, Heidelberg: Springer Berlin Heidelberg.

Czernik, Agnieszka (2016): Authentisierung, Authentifizierung und Autorisierung. Online verfügbar unter https://www.datenschutzbeauftragter-info.de/authentisierung-authentifizierung-und-autorisierung/, zuletzt aktualisiert am 24.06.2016, zuletzt geprüft am 14.12.2017.

disy Informationssysteme GmbH (2016): Cadenza | Disy Informationssysteme GmbH. disy Informationssysteme GmbH. Online verfügbar unter https://www.disy.net/produkte/cadenza.html, zuletzt geprüft am 06.06.2016.

Docker (2016): What is Docker. Online verfügbar unter https://www.docker.com/what-docker, zuletzt geprüft am 04.10.2016.

Domingue, John; Fensel, Dieter; Hendler, James A. (Hg.) (2011): Handbook of Semantic Web Technologies. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg (Cellular origin, life in extreme habitats and astrobiology, 19).

DuCharme, Bob (2014): Storing and querying RDF in Neo4j. bobdc.blog. Online verfügbar unter http://www.snee.com/bobdc.blog/2014/01/storing-and-querying-rdf-in-ne.html, zuletzt aktualisiert am 07.05.2015, zuletzt geprüft am 06.10.2016.

Düpmeier, Clemens (2012): SUI - Eine Service-orientierte Schnittstelle zur Einbindung von Fachsystemen in die semantische Suche nach Umweltinformationen. In: Kurt Weissenbach (Hg.): F+E-Vorhaben MAF-UIS, Moderne anwendungsorientierte Forschung und Entwicklung für Umweltinformationssysteme, Phase I 2011/12 : KIT Scientific Reports 7616. Karlsruhe: KIT.

Düpmeier, Clemens; Geiger, Werner; Greceanu, Claudia; Weidemann, Rainer; Ebel, Renate; Lehle, Manfred et al. (2008): Themenpark Umwelt - Erweiterung der Umwelt-kommunikations-Plattform um Fultimediafunktionalitäten und Inhalte des Bodensee-Webs. In: Roland Mayer-Föll, André Keitel und Werner Geiger (Hg.): KEWA Phase III, S. 77–84. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/servlet/is/91128/kewa3-gesamt-bildschirm.pdf?command=downloadContent&filename=kewa3-gesamt-bildschirm.pdf.

Düpmeier, Clemens; Geiger, Werner; Greceanu, Claudia; Weidemann, Rainer; Ebel, Renate; Lehle, Manfred et al. (2009): Themenpark Umwelt. Optimierung der Volltext-

suche und Integration von Panoramabildern und Live-Videos. In: Roland Mayer-Föll, André Keitel und Werner Geiger (Hg.): KEWA Phase IV - Kooperative Entwicklung wirtschaftlicher Anwendungen für Umwelt, Verkehr und benachbarte Bereiche in neuen Verwaltungsstrukturen, S. 167–174. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/servlet/is/93797/kewa4-bildschirm.pdf?command=downloadContent&filename=kewa4-bildschirm.pdf.

Düpmeier, Clemens; Geiger, Werner; Greceanu, Claudia; Weidemann, Rainer; Ruchter, Markus; Ebel, Renate et al. (2007): Themenpark Umwelt - Fortentwicklung des Themenparks Umwelt, Erprobung von Web 2.0 Technologien. In: Roland Mayer-Föll, André Keitel und Werner Geiger (Hg.): KEWA Phase II, S. 45–52. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/servlet/is/92260/kewa2-gesamt-bildschirm.pdf?command=downloadContent&filename=kewa2-gesamt-bildschirm.pdf.

Eberspächer, Jörg; Holtel, Stefan (2007): Suchen und Finden im Internet. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg. Online verfügbar unter http://dx.doi.org/10.1007/978-3-540-38157-0.

ECMA International (2013): The JSON Data Interchange Format. ECMA International. Online verfügbar unter http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf, zuletzt geprüft am 20.05.2017.

Edlich, Stefan (2011): NoSQL. Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken. 2., aktualisierte und erw. Aufl. München: Hanser. Online verfügbar unter http://www.hanser-elibrary.com/isbn/9783446427532.

Ehrig, Marc; Staab, Steffen; Sure, York (2005): Bootstrapping Ontology Alignment Methods with APFEL. In: *Proc. of the 4th Int. Semantic Web Conf. (ISWC-2005)*.

Elastic (2016): Elasticsearch. Elastic. Online verfügbar unter https://www.elastic.co/de/products/elasticsearch, zuletzt geprüft am 04.10.2016.

Eriksdotter, Holger (2009): Unstrukturierte Daten: Der ungehobene Schatz. Hg. v. Computerwoche. Online verfügbar unter https://www.computerwoche.de/a/der-ungehobene-schatz,1908337, zuletzt geprüft am 02.06.2016.

Erl, Thomas (2016): Service-oriented architecture. Concepts, technology, and design. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference (The Prentice Hall Service-oriented computing series from Thomas Erl).

Eugster, Patrick Th.; Felber, Pascal A.; Guerraoui, Rachid; Kermarrec, Anne-Marie (2003): The many faces of publish/subscribe. In: *ACM Comput. Surv.* 35 (2), S. 114–131. DOI: 10.1145/857076.857078.

European Environment Information and Observation Network (2017): GEMET - General Multilingual Environmental Thesaurus. Online verfügbar unter https://www.eionet.europa.eu/gemet/en/themes/, zuletzt geprüft am 03.10.2017.

Euzenat, Jérôme; Shvaiko, Pavel (2007): Ontology Matching. New York: Springer.

fancybox.net (2016): Fancybox - Fancy jQuery lightbox alternative. fancybox.net. Online verfügbar unter http://fancybox.net/, zuletzt geprüft am 02.06.2016.

Fielding, Roy Thomas (2000): Architectural Styles and the Design of Network-based Software Architectures. Dissertation. University of California, Irvine. Online verfügbar unter http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm, zuletzt geprüft am 20.06.2016.

Fleisch, Elgar; Mattern, Friedemann (Hg.) (2005): Das Internet der Dinge. Ubiquitous Computing und RFID in der Praxis: Visionen, Technologien, Anwendungen, Handlungsanleitungen ; mit 21 Tabellen. Berlin: Springer. Online verfügbar unter http://lib.myilibrary.com/detail.asp?id=62329.

Fowler, Martin (2006): Continuous Integration. Online verfügbar unter https://martinfowler.com/articles/continuousIntegration.html, zuletzt geprüft am 02.06.2016.

Fowler, Martin; Lewis, James (2015): Microservices: Nur ein weiteres Konzept in der Softwarearchitektur oder mehr? In: *OBJEKTspektrum* (01/2015).

Fraunhofer IOSB (2016): WebGenesis - Generierungssupport für Web-basierte Informationssysteme. Fraunhofer IOSB. Online verfügbar unter https://www.iosb.fraunhofer.de/servlet/is/18052/, zuletzt geprüft am 04.10.2016.

Fredrich, Todd (2013): RESTful Service Best Practices. Online verfügbar unter https://github.com/tfredrich/RestApiTutorial.com/blob/master/media/RESTful%20Best%20Practices-v1_2.pdf.

Freed, Ned; Borenstein, Nathaniel (1996): Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. Network Working Group. Online verfügbar unter https://tools.ietf.org/html/rfc2046, zuletzt aktualisiert am 01.11.1996, zuletzt geprüft am 14.12.2017.

Fröschle, Hans-Peter; Reinheimer, Stefan (Hg.) (2007): Serviceorientierte Architekturen. 1. Aufl. Heidelberg: dpunkt-Verl. (HMD - Praxis der Wirtschaftsinformatik, 44.2007,253). Online verfügbar unter http://deposit.d-nb.de/cgi-bin/dokserv?id=2894527&prov=M&dok_var=1&dok_ext=htm.

Gamma, Erich (2004): Design patterns. Elements of reusable object-oriented software. [Nachdr.]. Boston: Addison-Wesley (Addison-Wesley professional computing series).

Garrett, Jesse James (2005): Ajax: A New Approach to Web Applications. Online verfügbar unter https://web.archive.org/web/20080702075113/http://www.adaptivepath.com/ideas/essays/archives/000385.php, zuletzt aktualisiert am 18.02.2005, zuletzt geprüft am 12.07.2017.

Gliozzo, Alfio; Strapparava, Carlo (2009): Semantic Domains in Computational Linguistics. Berlin, Heidelberg: Springer Berlin Heidelberg.

Glöggler, Michael (2003): Suchmaschinen im Internet. Funktionsweisen, Ranking Methoden, Top Positionen. Berlin, Heidelberg: Springer Berlin Heidelberg (Xpert.press).

GlossarWiki (2017): Web-Portal. Hochschule Augsburg. Online verfügbar unter https://glossar.hs-augsburg.de/Web-Portal, zuletzt aktualisiert am 06.12.2017, zuletzt geprüft am 14.12.2017.

Google Inc. (2014a): KeyMatch. Google Inc. (Google Search Appliance Help Center). Online verfügbar unter https://www.google.com/support/enterprise/static/gsa/docs/admin/70/admin_console_help/serve_keymatch.html, zuletzt aktualisiert am 09.10.2014, zuletzt geprüft am 02.10.2017.

Google Inc. (2014b): Related Queries. Google Inc. (Google Search Appliance Help Center). Online verfügbar unter https://www.google.com/support/enterprise/static/gsa/docs/admin/70/admin_console_help/serve_synonym.html, zuletzt aktualisiert am 09.10.2014, zuletzt geprüft am 02.10.2017.

Google Inc. (2015): Google OneBox for Enterprise Developer's Guide. Online verfügbar unter https://www.google.com/support/enterprise/static/gsa/docs/admin/74/gsa_doc_set/oneboxguide/, zuletzt geprüft am 02.04.2017.

Greenfield, Adam (2006): Everyware. The dawning age of ubiquitous computing. Berkeley, CA: New Riders (Voices That Matter).

Griesbaum, Joachim; Rittberger, Marc; Bekavac, Bernard (2002): Deutsche Suchmaschinen im Vergleich: AltaVista.de, Fireball.de, Google.de und Lycos.de. In: *Proceedings des 8. Internationalen Symposiums für Informationswissenschaft*. Online verfügbar unter http://www.joachim-griesbaum.de/files/griesbaum_rittberger_bekavac.pdf, zuletzt geprüft am 20.10.2016.

Gründerszene Lexikon (2017): Agile Softwareentwicklung. Online verfügbar unter https://www.gruenderszene.de/lexikon/begriffe/agile-softwareentwicklung, zuletzt geprüft am 08.10.2017.

Gschwender, David; Kost, Florian; Schillinger, Wolfgang; Niemeier, Rüdiger; Koch, Lars; Düpmeier, Clemens; Schlachter, Thorsten (2016): Energieatlas Baden-Württemberg Daten und Fakten zur Energiewende. In: Kurt Weissenbach, Wolfgang Schillinger und Rainer Weidemann (Hg.): INOVUM Phase I 2014/16, S. 61–70. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/content/119257/INOVUM_I_Endfassung.pdf.

Guo, Chuanxiong; Lu, Guohan; Li, Dan; Wu, Haitao; Zhang, Xuan; Shi, Yunfeng et al. (2009): BCube. In: Pablo Rodriguez (Hg.): Proceedings of the ACM SIGCOMM 2009 conference on Data communication. the ACM SIGCOMM 2009 conference. Barcelona, Spain. ACM Special Interest Group on Data Communication. New York, NY: ACM, S. 63.

Hänsch, Carl-Philipp (2014): 8 Gründe, Daten stärker zu strukturieren. Online verfügbar unter https://launix.de/launix/8-gruende-daten-staerker-zu-strukturieren/, zuletzt aktualisiert am 02.09.2016, zuletzt geprüft am 08.10.2017.

Harold, Elliotte Rusty (2002): Die XML-Bibel. [XML-Einführung: alles zu Elementen, Tags, Attributen, DTD und Namensräumen ; beherrschen Sie die ganze Power von CSS und XSL ; reizen Sie die XML-Möglichkeiten voll aus mit XLinks, XPointer, Schemas, SVG und XHTML ; CD-ROM - XML-Browser und -Werkzeuge, XML W3C-Standards, Programmier-Code]. 2. und aktualis. Aufl. auf Grundlage von XML 1.0, 2. Ed. Bonn: mitp-Verl.

Harras, Gisela; Hass, Ulrike; Strauss, Gerhard (1991): Wortbedeutungen und ihre Darstellung im Wörterbuch. Berlin [Germany]: Walter de Gruyter & Co (Schriften des Instituts für deutsche Sprache, Band 3).

Heise online (2017): Wearables – Technik zum Tragen. Online verfügbar unter https://www.heise.de/thema/Wearables, zuletzt geprüft am 08.10.2017.

Herb, Ulrich (2012): Open Initiatives: Offenheit in der digitalen Welt und Wissenschaft: Universitätsverlag des Saarlandes. Online verfügbar unter http://universaar.uni-saar-land.de/monographien/volltexte/2012/87/pdf/Onlineversion_Open_Initiatives_Ulrich_Herb.pdf.

Heyer, Gerhard; Quasthoff, Uwe; Wittig, Thomas (2008): Text mining: Wissensrohstoff Text. Konzepte, Algorithmen, Ergebnisse. Korrigierter Nachdr. Herdecke: W3L-Verl. (Informatik). Online verfügbar unter http://deposit.ddb.de/cgi-bin/dokserv?id=2783785&prov=M&dok_var=1&dok_ext=htm.

Hickson, Ian; Berjon, Robin; Faulkner, Steve; Leithead, Travis; Navara, Erika Doyle; O'Connor, Edward; Pfeiffer, Silvia (2014): HTML5. A vocabulary and associated APIs for HTML and XHTML. W3C. Online verfügbar unter https://www.w3.org/TR/html5/, zuletzt aktualisiert am 24.10.2014, zuletzt geprüft am 02.06.2016.

Hitzler, Pascal; Markus, Krötzsch; Rudolph, Sebastian; York, Sure (2008): Semantic Web. Heidelberg: Springer.

Holten, Roland (1999): Semantische Spezifikation Dispositiver Informationsysteme. In: *Arbeitsberichte des Instituts für Wirtschaftsinformatik der Westfälischen Wilhelms-Universität Münster* (69).

Holzinger, Andreas (2014): Biomedical informatics. Discovering knowledge in big data. Cham: Springer.

Humble, Jez; Read, Chris; North, Dan (2006): The Deployment Production Line. In: Joseph Chao (Hg.): Agile Conference, 2006. 23 - 28 July 2006, [Minneapolis, Minnesota ; proceedings]. AGILE 2006 (AGILE'06). Minneapolis, MN, USA, 23-28 July 2006. Agile Alliance; Association for Computing Machinery; Agile Conference. Los Alamitos, Calif.: IEEE Computer Society, S. 113–118.

ISO/IEC (2006): Topic Maps - XML-Syntax. ISO/IEC. Online verfügbar unter http://www.isotopicmaps.org/sam/sam-xtm/, zuletzt geprüft am 04.10.2016.

Java Community Process (2008): JSR 286: Portlet Specification 2.0. Final Release. Java Community Process. Online verfügbar unter https://www.jcp.org/en/jsr/detail?id=286, zuletzt geprüft am 02.10.2017.

json-schema-org (2018): JSON Schema. Online verfügbar unter http://json-schema.org/, zuletzt aktualisiert am 12.03.2018, zuletzt geprüft am 03.04.2018.

Khadjeh Nassirtoussi, Arman; Aghabozorgi, Saeed; Ying Wah, Teh; Ngo, David Chek Ling (2014): Text mining for market prediction. A systematic review. In: *Expert Systems with Applications* 41 (16), S. 7653–7670. DOI: 10.1016/j.eswa.2014.06.009.

Koch, Wolfgang; Frees, Beate (2015): ARD/ZDF-Onlinestudie 2015. Unterwegsnutzung des Internets wächst bei geringerer Intensität. Hg. v. ARD/ZDF-Medienkommission. Online verfügbar unter http://www.ard-zdf-onlinestudie.de/files/2015/0915_Koch_Frees.pdf, zuletzt aktualisiert am 01.09.2015, zuletzt geprüft am 15.07.2016.

Krause, Florentin; Bossel, Hartmut; Müller-Reißmann, Karl F. (1981): Energie-Wende. Wachstum und Wohlstand ohne Erdöl und Uran ; ein Alternativ-Bericht des Öko-Instituts/Freiburg. 3. Aufl., 10.-12-Tsd. Frankfurt/M.: S. Fischer.

Kubernetes (2016): Kubernetes. Online verfügbar unter https://kubernetes.io/, zuletzt geprüft am 04.10.2016.

Lal, Rajesh; Chava, Lakshmi C. (2010): Developing Web Widget with HTML, CSS, JSON and AJAX. Scotts Valley, Calif.: Createspace.

Lang, Stefan M.; Lockemann, Peter C. (1995): Datenbankeinsatz. Berlin, Heidelberg: Springer Berlin Heidelberg; Imprint; Springer.

Lange, Jürgen (2009): Datenflut – Fluch oder Segen? Wie Sie mit Enterprise Search einfach und sicher Informationen finden. Frankfurt: Frankfurter Allgemeine Buch.

Lemke, Matthias; Wiedemann, Gregor; Blätte, Andreas (2016): Text Mining in den Sozialwissenschaften. Grundlagen und Anwendungen zwischen qualitativer und quantitativer Diskursanalyse.

Leong, Lydia; Bala, Raj; Lowery, Craig; Smith, Dennis (2017): Magic Quadrant for Cloud Infrastructure as a Service, Worldwide. Gartner. Online verfügbar unter https://www.gartner.com/doc/reprints?id=1-2G2O5FC&ct=150519, zuletzt aktualisiert am 15.06.2017, zuletzt geprüft am 25.07.2017.

Lewandoski, Dirk; Höchstötter, Nadine (2007): Qualitätsmessung bei Suchmaschinen. In: *Informatik-Spektrum* 30 (3), S. 159–169.

Lewandowski, Dirk (2015): Suchmaschinen verstehen. Berlin Heidelberg: Springer Vieweg.

Lewandowski, Dirk; Mayr, Philipp (2006): Exploring the Academic Invisible Web. In: *Library Hi Tech* 24(4).

Liferay (2014): Liferay. Liferay. Online verfügbar unter https://www.liferay.com/de/home, zuletzt geprüft am 01.05.2014.

Liferay (2016): Inter-portlet communication. Liferay GmbH. Online verfügbar unter https://web.liferay.com/de/community/wiki/-/wiki/Main/Inter-portlet+communication, zuletzt geprüft am 10.10.2016.

LUBW (2016): Daten- und Kartendienst der LUBW. Online verfügbar unter http://udo.lubw.baden-wuerttemberg.de/public/, zuletzt geprüft am 06.06.2016.

Lunapark (2015): Suchmaschinenmarktanteile weltweit 2015.

Mangold, Christoph (2007): A survey and classification of semantic search approaches. In: *IJMSO* 2 (1), S. 23. DOI: 10.1504/IJMSO.2007.015073.

material.io (2017): Material Design - Introduction. Online verfügbar unter https://material.io/guidelines/, zuletzt geprüft am 08.10.2017.

Mayer-Föll, Roland (1992): Zur Rahmenkonzeption Des Umweltinformationssystems Baden-Württemberg. In: Oliver Günther, Franz Josef Radermacher, Helmut Kuhn und Roland Mayer-Föll (Hg.): Konzeption und Einsatz von Umweltinformationssystemen. Proceedings, Bd. 301. Berlin, Heidelberg: Springer (Informatik-Fachberichte, 301), S. 3–19.

Mayer-Föll, Roland (1993): Das Umweltinformationssystem Baden-Württemberg Zielsetzung und Stand der Realisierung. In: Andreas Jaeschke, Thomas Kämpke, Bernd Page und Franz Josef. J. Radermacher (Hg.): Informatik für den Umweltschutz. 7. Symposium, Ulm, 31.3.-2.4.1993. Berlin, Heidelberg: Springer (Informatik aktuell), S. 313–337.

McKay, Dana (2011): Gotta keep 'em separated. In: Sally Jo Cunningham (Hg.): Proceedings of the 12th Annual Conference of the New Zealand Chapter of the ACM Special Interest Group on Computer-Human Interaction. the 12th Annual Conference of the New Zealand Chapter of the ACM Special Interest Group. Hamilton, New Zealand, 4/7/2011 - 5/7/2011. New York, NY: ACM, S. 109–112.

memosens.org (2017): Memosens - De-facto-Standard in der Prozessanalytik. Online verfügbar unter http://www.memosens.org/hersteller.html, zuletzt geprüft am 17.07.2017.

Mesbah, Ali; van Deursen, Arie (2007): Migrating Multi-page Web Applications to Single-page AJAX Interfaces. In: René Krikhaar (Hg.): 11th European Conference on Software Maintenance and Reengineering, 2007. CSMR '07 ; 21 - 23 March 2007, Amsterdam, the Netherlands. 11th European Conference on Software Maintenance and Reengineering (CSMR'07). Amsterdam, The Netherlands. Reengineering Forum Industry Association; IEEE Computer Society; European Conference on Software Mainte-

nance and Reengineering; CSMR. Los Alamitos, Calif.: IEEE Computer Society, S. 181–190.

Mikowski, Michael S.; Powell, Josh C.; Benson, Gregory D. (2014): Single page web applications. Javascript end-to-end. Shelter Island, NY: Manning.

Mitra, Nilo; Lafon, Yves (2007): SOAP Version 1.2. Part 0: Primer (Second Edition). W3C. Online verfügbar unter https://www.w3.org/TR/2007/REC-soap12-part0-20070427/, zuletzt aktualisiert am 27.04.2007, zuletzt geprüft am 30.08.2017.

Mouat, Adrian (2016): Docker. Software entwickeln und deployen mit Containern. Heidelberg: dpunkt.verlag. Online verfügbar unter http://gbv.eblib.com/patron/FullRecord.aspx?p=4712032.

MSS Alliance (2015): MSS Alliance Launched to Set De Facto Standard for Odor-Sensing Systems. Online verfügbar unter https://www.nec.com/en/press/201510/global_20151013_05.html, zuletzt aktualisiert am 13.10.2015, zuletzt geprüft am 17.07.2017.

Naiman, Channah F.; Ouksel, Arison M. (1995): A classification of semantic conflicts in heterogeneous database systems. In: *Journal of Organizational Computing* 5 (2), S. 167–193. DOI: 10.1080/10919399509540248.

neo4j (2016): Neo4j, the world's leading graph database - Neo4j Graph Database. Neo4j, Inc. Online verfügbar unter https://neo4j.com/, zuletzt geprüft am 06.10.2016.

Nikolai, Ralf (2002): Thesaurusföderationen: Ein Rahmenwerk für die flexible Integration von heterogenen, autonomen Thesauri. Dissertation. Universität Karlsruhe, Karlsruhe. Online verfügbar unter https://publikationen.bibliothek.kit.edu/4562003/3208, zuletzt geprüft am 04.10.2016.

Noy, Natasha (2009): Ontology Mapping. Hg. v. S. Staab und R. Studer. Berlin, Heidelberg: Springer (International handbooks on information systems. Handbook on onologies).

OASIS Open (2006): Reference Model for Service Oriented Architecture 1.0. OASIS Open. Online verfügbar unter https://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf, zuletzt geprüft am 31.07.2017.

OASIS UDDI Specifications TC (2016): OASIS - Committees - OASIS UDDI Specifications TC. OASIS Open. Online verfügbar unter https://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm, zuletzt geprüft am 02.06.2016.

Olliffe, Gary (2015): Microservices : Building Services with the Guts on the Outside. Gartner. Online verfügbar unter http://blogs.gartner.com/gary-olliffe/2015/01/30/microservices-guts-on-the-outside/, zuletzt geprüft am 04.10.2016.

opensearch.org (2013): OpenSearch Specifications 1.1/Draft 5. Online verfügbar unter http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch_description_document, zuletzt aktualisiert am 17.06.2013, zuletzt geprüft am 04.10.2016.

Pellegrini, Tassilo; Blumauer, Andreas (2006): Semantic Web. Wege zur vernetzten Wissensgesellschaft. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg (X.media.press). Online verfügbar unter http://dx.doi.org/10.1007/3-540-29325-6.

Ray, Eric T. (2002): Einführung in XML. 1. Aufl., korrigierter Nachdr. Beijing: O'Reilly.

React (2017): React - A JavaScript library for building user interfaces. Online verfügbar unter https://reactjs.org/, zuletzt geprüft am 08.10.2017.

Revolvy (2016): List of subject-predicate-object databases. Revolvy.com. Online verfügbar unter https://www.revolvy.com/main/index.php?s=List%20of%20subject-predicate-object%20databases, zuletzt geprüft am 06.10.2016.

Richardson, Leonard (2007): Web-Services mit REST. Title from resource description page (viewed on April 27, 2009). - Includes index. 1. Aufl. Köln: O'Reilly Verlag (Safari Books Online). Online verfügbar unter http://proquest.safaribooksonline.com/9783897217270.

Ritchie, Colin (2002): Relational database principles. 2nd ed. London: Continuum.

Rüther, Maria; Bandholtz, Thomas (2008): 5 Jahre Semantic Network Service (SNS) Aktueller Stand und Ausblick. In: Gerlinde Knetsch, Karin Jessen, Ines Beckmann (Hg.): Workshop des Arbeitskreises "Umweltdatenbanken / Umweltinformationssysteme“: Umweltbundesamt, S. 69–88.

schema.org (2016a): About schema.org. schema.org. Online verfügbar unter http://schema.org/docs/about.html, zuletzt geprüft am 02.06.2016.

schema.org (2016b): About Schema.org. schema.org. Online verfügbar unter http://schema.org/docs/about.html, zuletzt geprüft am 02.06.2016.

schema.org (2016c): Getting started with schema.org using Microdata. schema.org. Online verfügbar unter https://schema.org/docs/gs.html, zuletzt geprüft am 08.10.2017.

Schlachter, Thorsten; Düpmeier, Clemens; Geiger, Werner; Weidemann, Rainer; Ebel, Renate; Tauber, Martina et al. (2011a): Concept of a universal mobile application accessing environmental information systems. In: *Innovations in Sharing Environmental Observations and Information : EnviroInfo 2011 : 25th Internat.Conf.on Environmental Informatics*, S. 398–504.

Schlachter, Thorsten; Düpmeier, Clemens; Greceanu, Claudia; Weidemann, Rainer; Weissenbach, Kurt; Rossi, R. et al. (2014a): Cloud-Dienste - Erste Ergebnisse der Evaluierung von Cloud-Diensten für das UIS Baden-Württemberg. In: *KIT SCIENTIFIC REPORTS* 2014 (7665), S. 35–44. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/servlet/is/112361/13-iai-clouddienste.pdf?command=downloadContent&filename=13-iai-clouddienste.pdf.

Schlachter, Thorsten; Düpmeier, Clemens; Weidemann, Rainer; Schillinger, Wolfgang; Bayer, Nina; Hrebicek, J. (2013): 'My environment' - a dashboard for environmental information on mobile devices. In: *Environmental Software Systems : Fostering Infor-*

mation Sharing; 10th IFIP WG 5.11 International Symposium. (ISESS 2013), S. 197–203.

Schlachter, Thorsten; Geiger, Werner; Weidemann, Rainer; Zilly, Gerd; Ebel, Renate; Tauber, Martina et al. (2008): Landes-Umweltportale - Vernetzung von Informationen in den Umweltportalen von Baden-Württemberg, Sachsen-Anhalt und Thüringen unter Einsatz einer kommerziellen Suchmaschine. In: Roland Mayer-Föll, André Keitel und Werner Geiger (Hg.): Forschungszentrum Karlsruhe, Wissenschaftliche Berichte. Karlsruhe: Forschungszentrum Karlsruhe, S. 63–76.

Schlachter, Thorsten; Geiger, Werner; Weidemann, Rainer; Zilly, Gerd; Ebel, Renate; Tauber, Martina et al. (2011b): LUPO - Bereitstellung flexibel nutzbarer Dienste in Landesumweltportalen. In: Roland Mayer-Föll, Renate Ebel und Werner Geiger (Hg.): KEWA Phase VI, S. 9–20. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/servlet/is/100264/kewa6-iai-lupo.pdf?command=downloadContent&filename=kewa6-iai-lupo.pdf.

Schlachter, Thorsten; Greceanu, Claudia; Düpmeier, Clemens; Schmitt, Christian; Weidemann, Rainer; Schillinger, Wolfgang et al. (2014b): LUPO. Weiterentwicklung der Landesumweltportale, Karlsruhe. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/servlet/is/112375/16-iai-lupo.pdf?command=downloadContent&filename=16-iai-lupo.pdf.

Schlachter, Thorsten; Weidemann, Rainer; Ebel, Renate; Schillinger, Wolfgang; Zetzmann, Klaus (2012): Building Mobile Environmental Apps Using Web and Cloud Technologies. In: Hans-Knut Arndt (Hg.): 26th International Conference on Informatics for Environmental Protection (EnviroInfo 2012) : Proc.of the 26th Internat.Conf.on Informatics for Environmental Protection, Bd. 1: Shaker, S. 385–392.

Schulte, W. Roy; Natis, Yefim V. (1996): "Service Oriented" Architectures. Gartner. Online verfügbar unter https://www.gartner.com/doc/302868.

Schurman, Eric; Brutlag, Jake (2009): The User and Business Impact of Server Delays, Additional Bytes, and HTTP Chunking in Web Search. Performance Related Changes and their User Impact. Online verfügbar unter https://cdn.oreillystatic.com/en/assets/1/event/29/The%20User%20and%20Business%20Impact%20of%20Server%20Delays%2C%20Additional%20Bytes%2C%20and%20HTTP%20Chunking%20in%20Web%20Search%20Presentation.pptx, zuletzt aktualisiert am 23.06.2009, zuletzt geprüft am 13.12.2017.

Service-BW (2016): Hilfe in allen Lebenslagen - Serviceportal Baden-Württemberg. Service-BW. Online verfügbar unter https://www.service-bw.de/lebenslagen, zuletzt geprüft am 04.10.2016.

Sherman, Chris; Price, Gary (2001): The invisible web: uncovering sources search engines can't see. Hg. v. University of Illinois at Urbana-Champaign.

Shirky, Clay (2005): Ontology is Overrated. Online verfügbar unter http://shirky.com/writings/herecomeseverybody/ontology_overrated.html, zuletzt geprüft am 01.06.2018.

Siemens (2014): Internet der Dinge. Online verfügbar unter https://www.siemens.com/innovation/de/home/pictures-of-the-future/digitalisierung-und-software/internet-der-dinge-eingebettete-systeme.html, zuletzt aktualisiert am 01.10.2014, zuletzt geprüft am 08.10.2017.

Sikos, Leslie F. (2015): Mastering structured data on the Semantic Web. From HTML5 microdata to linked open data. [Berkeley, CA]: Apress (The expert's voice in web development).

Sourceforge.net (2016): The OWL API. Online verfügbar unter http://owlcs.github.io/owlapi/, zuletzt geprüft am 08.10.2017.

Sowa, John F. (2014): Principles of Semantic Networks. Explorations in the Representation of Knowledge. Online-Ausg. Burlington: Elsevier Science (EBL-Schweitzer).

Stock, Wolfgang G.; Stock, Mechtild (2008): Wissensrepräsentation. Informationen auswerten und bereitstellen: De Gruyter Oldenbourg. Online verfügbar unter http://www.degruyter.com/search?f_0=isbnissn&q_0=9783486844900&searchTitles=true.

Sullivan, Dan (2015): NoSQL for mere mortals. Upper Saddle River, NJ: Pearson Education (For mere mortals series). Online verfügbar unter http://proquest.tech.safaribooksonline.de/9780134029894, zuletzt geprüft am 01.06.2018.

Swartz, Aaron (2013): A Programmable Web. An Unfinished Work. In: *Synthesis Lectures on the Semantic Web: Theory and Technology* 3 (2), S. 1–64. DOI: 10.2200/S00481ED1V01Y201302WBE005.

TopicMaps.Org (2001): XML Topic Maps (XTM) 1.0. TopicMaps.Org. Online verfügbar unter http://topicmaps.org/xtm/, zuletzt geprüft am 04.10.2016.

Trapp, Tobias (2007): Web Services: Overhead of SOAP Runtime. SAP Blogs. Online verfügbar unter https://blogs.sap.com/2007/05/04/web-services-overhead-of-soap-runtime/, zuletzt geprüft am 02.06.2016.

Typo3 (2014): TYPO3 - The Enterprise Open Source CMS. Typo3. Online verfügbar unter https://typo3.org/, zuletzt geprüft am 04.10.2016.

Uckelmann, Dieter; Harrison, Mark; Michahelles, Florian (2011a): An Architectural Approach Towards the Future Internet of Things. In: Dieter Uckelmann, Mark Harrison und Florian Michahelles (Hg.): Architecting the Internet of Things. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 1–24. Online verfügbar unter https://doi.org/10.1007/978-3-642-19157-2_1, zuletzt geprüft am 01.06.2018.

Uckelmann, Dieter; Harrison, Mark; Michahelles, Florian (2011b): Architecting the Internet of Things. Berlin: Springer-Verlag.

Umweltbundesamt (2016): Semantischer Netzwerk Service (SNS). Das gemeinsame Wortgut der Umweltinformation und die Umweltchronik. Umweltbundesamt. Online verfügbar unter https://sns.uba.de/de, zuletzt geprüft am 02.06.2016.

Urban Airship (2016): Push Notifications Explained. Online verfügbar unter https://www.urbanairship.com/push-notifications-explained, zuletzt geprüft am 08.10.2016.

W3C (1999): HTTP/1.1: Content Negotiation. W3C. Online verfügbar unter https://www.w3.org/Protocols/rfc2616/rfc2616-sec12.html, zuletzt aktualisiert am 01.09.2004, zuletzt geprüft am 04.10.2016.

W3C (2001): Web Service Definition Language (WSDL). W3C. Online verfügbar unter https://www.w3.org/TR/wsdl.html, zuletzt aktualisiert am 14.03.2001, zuletzt geprüft am 02.06.2016.

W3C (2004a): OWL Web Ontology Language Semantics and Abstract Syntax. W3C. Online verfügbar unter https://www.w3.org/TR/owl-semantics/, zuletzt aktualisiert am 02.10.2017, zuletzt geprüft am 02.10.2017.

W3C (2004b): RDF Vocabulary Description Language 1.0: RDF Schema. Online verfügbar unter https://www.w3.org/2001/sw/RDFCore/Schema/200203/, zuletzt aktualisiert am 30.04.2002, zuletzt geprüft am 08.10.2017.

W3C (2004c): Resource Description Framework (RDF). W3C. Online verfügbar unter https://www.w3.org/RDF/, zuletzt geprüft am 04.10.2016.

W3C (2004d): SOAP Specifications. Latest SOAP versions. W3C, zuletzt aktualisiert am 05.06.2007, zuletzt geprüft am 30.08.2017.

W3C (2007): Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C. Online verfügbar unter https://www.w3.org/TR/wsdl20/, zuletzt geprüft am 02.06.2016.

W3C (2008): SPARQL Query Language for RDF. Online verfügbar unter https://www.w3.org/TR/rdf-sparql-query/, zuletzt aktualisiert am 15.01.2008, zuletzt geprüft am 08.10.2017.

W3C (2009a): Same Origin Policy. W3C. Online verfügbar unter https://www.w3.org/Security/wiki/Same_Origin_Policy, zuletzt geprüft am 10.10.2016.

W3C (2009b): SKOS Simple Knowledge Organization System. In: Miles Alistair und Sean Bechhofer (Hg.): Skos simple knowledge organization system - reference.

W3C (2009c): Skos simple knowledge organization system - reference. Hg. v. Miles Alistair und Sean Bechhofer. Online verfügbar unter http://www.w3.org/TR/skos-reference, zuletzt geprüft am 02.04.2017.

W3C (2012a): OWL 2 Web Ontology Language Document Overview (Second Edition). W3C. Online verfügbar unter https://www.w3.org/TR/owl2-overview/, zuletzt geprüft am 04.10.2016.

W3C (2012b): XML Schema. Online verfügbar unter https://www.w3.org/standards/xml/schema, zuletzt aktualisiert am 20.07.2016, zuletzt geprüft am 03.04.2018.

W3C (2016): Web Components Current Status - W3C. W3C. Online verfügbar unter https://www.w3.org/standards/techs/components#w3c_all, zuletzt aktualisiert am 13.10.2016, zuletzt geprüft am 23.11.2016.

W3C OWL Working Group (2012): OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation 11 December 2012. Online verfügbar unter https://www.w3.org/TR/2012/REC-owl2-overview-20121211/, zuletzt aktualisiert am 09.12.2012, zuletzt geprüft am 30.08.2017.

Wache, Holger (2003): Semantische Mediation für heterogene Informationsquellen. Zugl.: Bremen, Univ., Diss., 2002. Berlin: Akad. Verl.-Ges. Aka (Dissertationen zur künstlichen Intelligenz, 261).

Wandiger, Peer (2009): Warum dominiert Google? Die Erfolgsgeheimnisse von Google! Online verfügbar unter https://www.selbstaendig-im-netz.de/google/warum-dominiert-google-die-erfolgsgeheimnisse-von-google/, zuletzt aktualisiert am 01.04.2009, zuletzt geprüft am 08.10.2017.

Wang, Ting; Su, Zhiyang; Xia, Yu; Muppala, Jogesh; Hamdi, Mounir (2015): Designing efficient high performance server-centric data center network architecture. In: *Computer Networks* 79, S. 283–296. DOI: 10.1016/j.comnet.2015.01.006.

Wang, Ting; Xia, Yu; Lin, Dong; Hamdi, Mounir (2014): Improving the efficiency of server-centric data center network architectures. In: Abbas Jamalipour (Hg.): IEEE International Conference on Communications (ICC), 2014. 10 - 14 June 2014, Sydney, Australia. ICC 2014 - 2014 IEEE International Conference on Communications. Sydney, NSW. Institute of Electrical and Electronics Engineers; IEEE International Conference on Communications; IEEE ICC. Piscataway, NJ: IEEE, S. 3088–3093.

webcomponents.org (2016): Web Components - Introduction. webcomponents.org. Online verfügbar unter https://www.webcomponents.org/introduction, zuletzt geprüft am 04.10.2016.

Weber, Rolf H.; Weber, Romana (2010): Internet of things. Legal perspectives. License ed. Berlin: Springer (Publikationen aus dem Zentrum für Informations- und Kommunikationsrecht der Universität Zürich, 49).

Weidemann, Rainer; Geiger, Werner; Greceanu, Claudia; Schlachter, Thorsten; Zilly, Gerd; Lautner, Petra et al. (2007): FADO BW - Realisierung erster Komponenten für ein verteiltes Fachdokumentenmanagement im Umweltinformationssystem Baden-Württemberg. In: Roland Mayer-Föll, André Keitel und Werner Geiger (Hg.): KEWA Phase II, S. 31–44.

Weidemann, Rainer; Geiger, Werner; Greceanu, Claudia; Schlachter, Thorsten; Zilly, Gerd; Lautner, Petra et al. (2008): FADO BW - Entwicklung der Basisversion für das

neue Fachdokumentenmanagement im Umweltinformationssystem Baden-Württemberg. In: Roland Mayer-Föll, André Keitel und Werner Geiger (Hg.): KEWA Phase III, S. 85–98. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/servlet/is/91128/kewa3-gesamt-bildschirm.pdf?command=downloadContent&filename=kewa3-gesamt-bildschirm.pdf.

Weidemann, Rainer; Geiger, Werner; Greceanu, Claudia; Schlachter, Thorsten; Zilly, Gerd; Lautner, Petra et al. (2009): FADO – Ablösung der XfaWeb-Systeme durch Fachdokumente Online, das neue Fachdokumentenmanagement im Umweltinformationssystem Baden-Württemberg. In: Roland Mayer-Föll, André Keitel und Werner Geiger (Hg.): KEWA Phase IV - Kooperative Entwicklung wirtschaftlicher Anwendungen für Umwelt, Verkehr und benachbarte Bereiche in neuen Verwaltungsstrukturen, S. 175–184. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/content/93797/kewa4-bildschirm.pdf.

Weidemann, Rainer; Geiger, Werner; Schlachter, Thorsten; Zilly, Gerd; Ebel, Renate; Hahn, R. et al. (2010): FADO – Funktionale Konsolidierung des Fachdokumentenmanagements im Umwelt- informationssystem Baden-Württemberg und Erschließung neuer Themenbereiche. In: Roland Mayer-Föll, Renate Ebel und Werner Geiger (Hg.): Umweltinformationssystem Baden-Württemberg. F+E-Vorhaben KEWA; Phase V, 2009/10. Karlsruhe, Baden: Universität Karlsruhe Universitätsbibliothek (KIT Scientific Reports, 7544), S. 75–84. Online verfügbar unter http://www.fachdokumente.lubw.baden-wuerttemberg.de/content/96419/kewa5-Bildschirm.pdf.

Weinhardt, Christof; Anandasivam, Arun; Blau, Benjamin; Borissov, Nikolay; Meinl, Thomas; Michalk, Wibke; Stößer, Jochen (2009): Cloud Computing – A Classification, Business Models, and Research Directions. In: *Bus. Inf. Syst. Eng.* 1 (5), S. 391–399. DOI: 10.1007/s12599-009-0071-2.

Wiegand, Stepanie (2013): And Now for Something Completely Different: Using OWL with Neo4j - Neo4j Graph Database. Online verfügbar unter https://neo4j.com/blog/using-owl-with-neo4j/, zuletzt aktualisiert am 21.08.2013, zuletzt geprüft am 06.10.2016.

Wikipedia (2016): HATEOAS - Wikipedia. Hg. v. Wikipedia. Online verfügbar unter https://en.wikipedia.org/wiki/HATEOAS, zuletzt geprüft am 02.06.2016.

Witte, René; Mülle, Jutta (Hg.) (2006): Text Mining. Wissensgewinnung aus natürlichsprachigen Dokumenten. Universität Karlsruhe, Fakultät für Informatik, Institut für Programmstrukturen und Datenorganisation (IPD) (Interner Bericht 2006-5).

Wolfram|Alpha (2017): Wolfram|Alpha: Making the world’s knowledge computable. Online verfügbar unter https://www.wolframalpha.com/, zuletzt geprüft am 02.04.2017.

WorldWideWebSize.com (2017): The size of the World Wide Web. Online verfügbar unter http://www.worldwidewebsize.com/, zuletzt geprüft am 03.04.2018.

Wrigth, Alex (2009): Exploring a ‘Deep Web’ That Google Can’t Grasp. In: *The New York Times*. Online verfügbar unter https://www.nytimes.com/2009/02/23/technology/23iht-23search.20357326.html, zuletzt geprüft am 08.10.2017.