



## Gemeinsamer Sachbericht

# LIMBO - Linked Data Services for Mobility



### Förderkennzeichen aller Projektpartner im Verbundprojekt:

eccenca GmbH (19F2029A), CISS TDI GmbH (19F2029B), Universität Paderborn (19F2029C), YellowMap AG (19F2029D), USU Software AG (19F2029E), appPlant GmbH (19F2029F), Institut für Angewandte Informatik e.V. (19F2029G), TU Berlin (19F2029H), Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme (19F2029I)

### Gefördert durch:



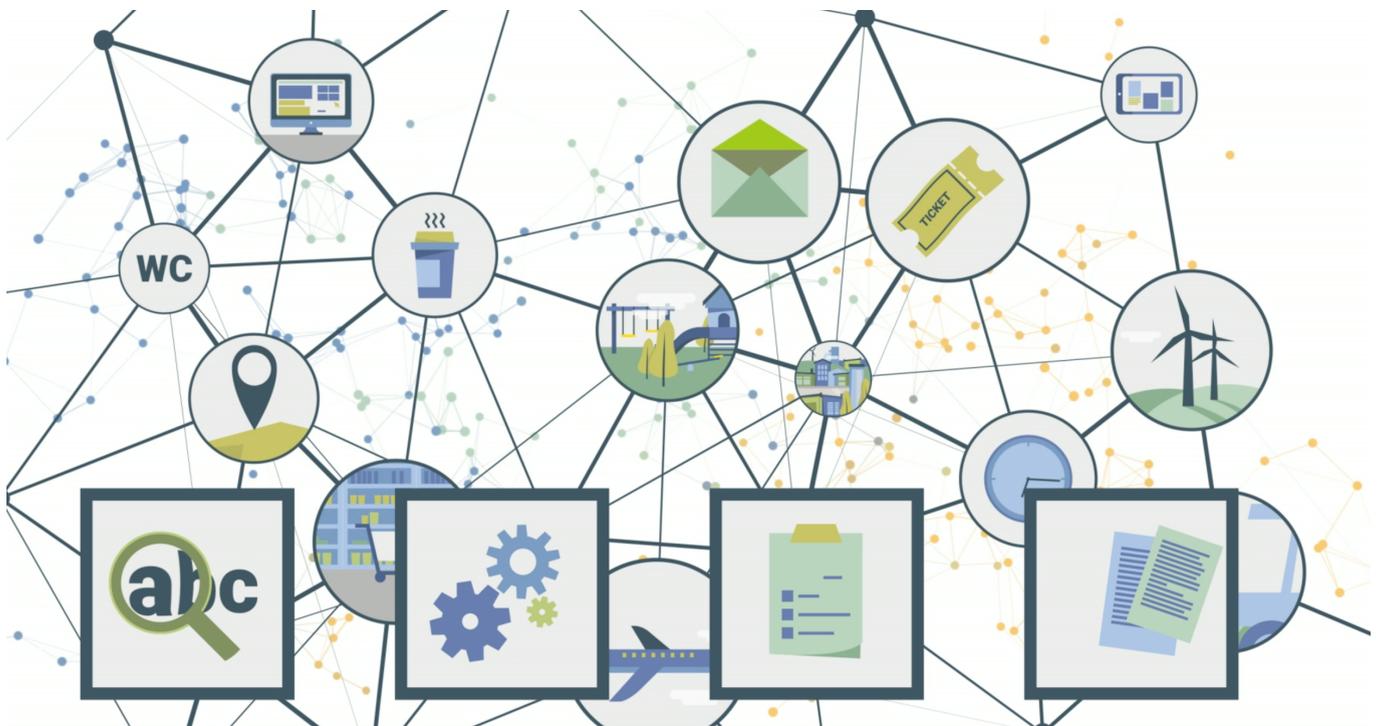
## Inhaltsverzeichnis

<b>Einleitung</b>	<b>3</b>
<b>Dienste und Werkzeuge</b>	<b>4</b>
Lifting-Portal	4
RDFTranslator	9
Quality Assurance Suite - QuAC	10
OGC-konforme Bereitstellung von Geodaten durch ein Data Warehouse (GDW)	14
Stichwortsuche und Question Answering	19
LIMBO-Pipeline	20
Metadaten-Management	22
Orientierungs-App(s) für Sehgeschädigte	24
VoCoL	26
<b>Anwendungsszenarien</b>	<b>28</b>
Suche auf Mobilitätsdaten	28
Weißflächenkartierung	28
Intelligente Service-Unterstützung (Industrie 4.0)	34
SmartCharging-Use-Case	35
Standortplanung für Infrastrukturen	38
<b>Daten und Ontologien</b>	<b>40</b>
CKAN-Portal	40
ALKIS	40
COSMO-REA	41
Industrie 4.0	41
DWD-Wetter Ontology	41
MobiVoc	41
Extraktion aus unstrukturierten Daten	42
Linking	43
Fusion	44
<b>Vernetzung</b>	<b>45</b>
<b>Publikationen</b>	<b>46</b>
2020	46
2019	46
2018	47
2017	48

## Einleitung

In offenen frei zugänglichen Daten (*Open Data*) im Allgemeinen und im Besonderen in mobilitätsrelevanten Daten schlummert ein enormes Potential. Die Open-Data-Angebote in Deutschland sind über viele Portale verteilt, sowie heterogen strukturiert und beschrieben. Mit der mCLOUD und dem MDM stellt das BMVI für mobilitätsrelevante Daten ein bundesweites infrastrukturelles Angebot für Datenbereitsteller und Dateninteressenten bereit.

Das zentrale Ziel des Projektes LIMBO bestand in der Optimierung der Datenqualität unter Aspekten der (a) Suche relevanter Informationen, (b) maschinenlesbaren Weiterverarbeitung und (c) nachhaltigen Bereitstellung, und in der Konsequenz einem stark erleichterten, barriere-ärmeren Zugang. Durch einen *Mobility Data Space* wurden durch Limbo bereitgestellte offene Daten des Bundesministeriums für Verkehrs & digitale Infrastruktur und administrativer Datenbereitsteller einer breiten Öffentlichkeit unter genannten Aspekten zur Verfügung gestellt. LIMBO vereinheitlichte und veredelte die Daten und schaffte so die Grundlage für viele neue, innovative Apps und Anwendungen.



Umgesetzt wurde das Vorhaben mithilfe semantischer Technologien. Durch sie werden die verschiedenen Datenquellen und Datensätze verknüpft und inhaltlich (semantisch) beschrieben. Darüber hinaus werden Daten des Mobility Data Space mit anderen offenen Semantic-Web-Datenquellen vernetzt. Relevante Informationen können somit effizienter gefunden und unmittelbar angewendet werden.

Die LIMBO-Ergebnisse können daher wie folgt zusammengefasst werden:

- **Vereinfachter Zugriff** auf die Daten der mCLOUD des BMVI.
- **Vereinfachte Integration** von Daten des Semantic Web.
- **Einheitliche Darstellung** öffentlicher und nicht-öffentlicher Daten.
- **Neue Dienste** zur Verarbeitung, Qualitätssicherung, Veredelung und Durchsuchung von Daten.
- **Grundlagen für innovative Anwendungsfälle** im Kontext der Mobilität 4.0 und Industrie 4.0.

Dieses Dokument strukturiert die Ergebnisse des Projektes nach **Diensten und Werkzeugen**, **Anwendungsszenarien** sowie **Daten und Ontologien**. Eine Auflistung aller **Kooperationen** und der im Projekt entstandenen **Publikationen** schließt diesen Bericht ab.

## Dienste und Werkzeuge

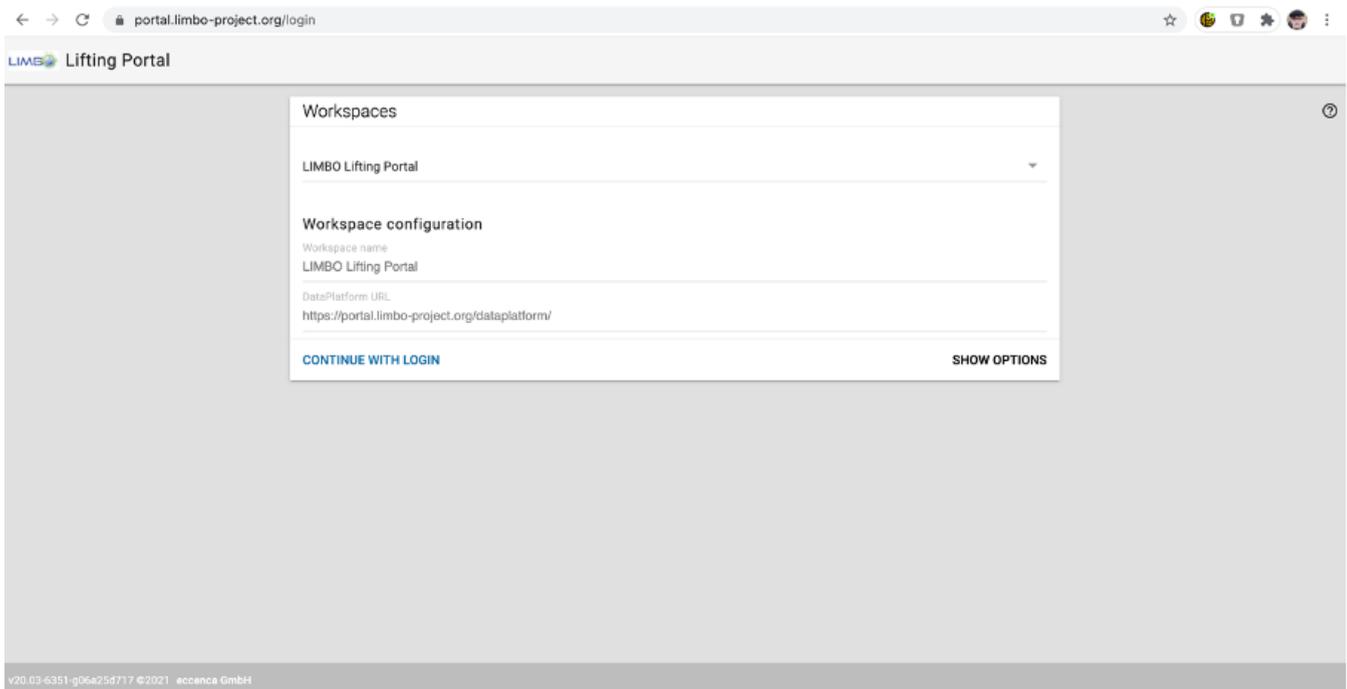
### Lifting-Portal

Das für das LIMBO-Projekt zentrale Lifting Portal wurde von eccenca entwickelt und bereitgestellt. Es basiert auf Forschungsergebnissen und Methoden aus den Arbeitspaketen AP3 (Vokabular Management, Authoring, Qualität), AP4 (Akquise, Discovery, Extraktion, Mapping), AP5 (Integration und Fusion) und AP6 (Exploration, Export, Bereitstellung) um in der Konsequenz hochqualitative Daten bereitzustellen.

Das Lifting-Portal wurde während der Projektlaufzeit als Service mit Anmeldung für alle Projektpartner bereitgestellt, um gemeinsam offene und vernetzte Datenbestände bereitstellen zu können.

Das Lifting-Portal konzentriert sich insbesondere auf die Funktionalitäten, welche notwendig sind, um strukturierte Roh-Daten (nicht zwingend RDF-Daten) aus Open Data Portalen und unternehmensinternen Quellen (Data-Lake) mit Hilfe von semantischen Technologien zu integrieren.

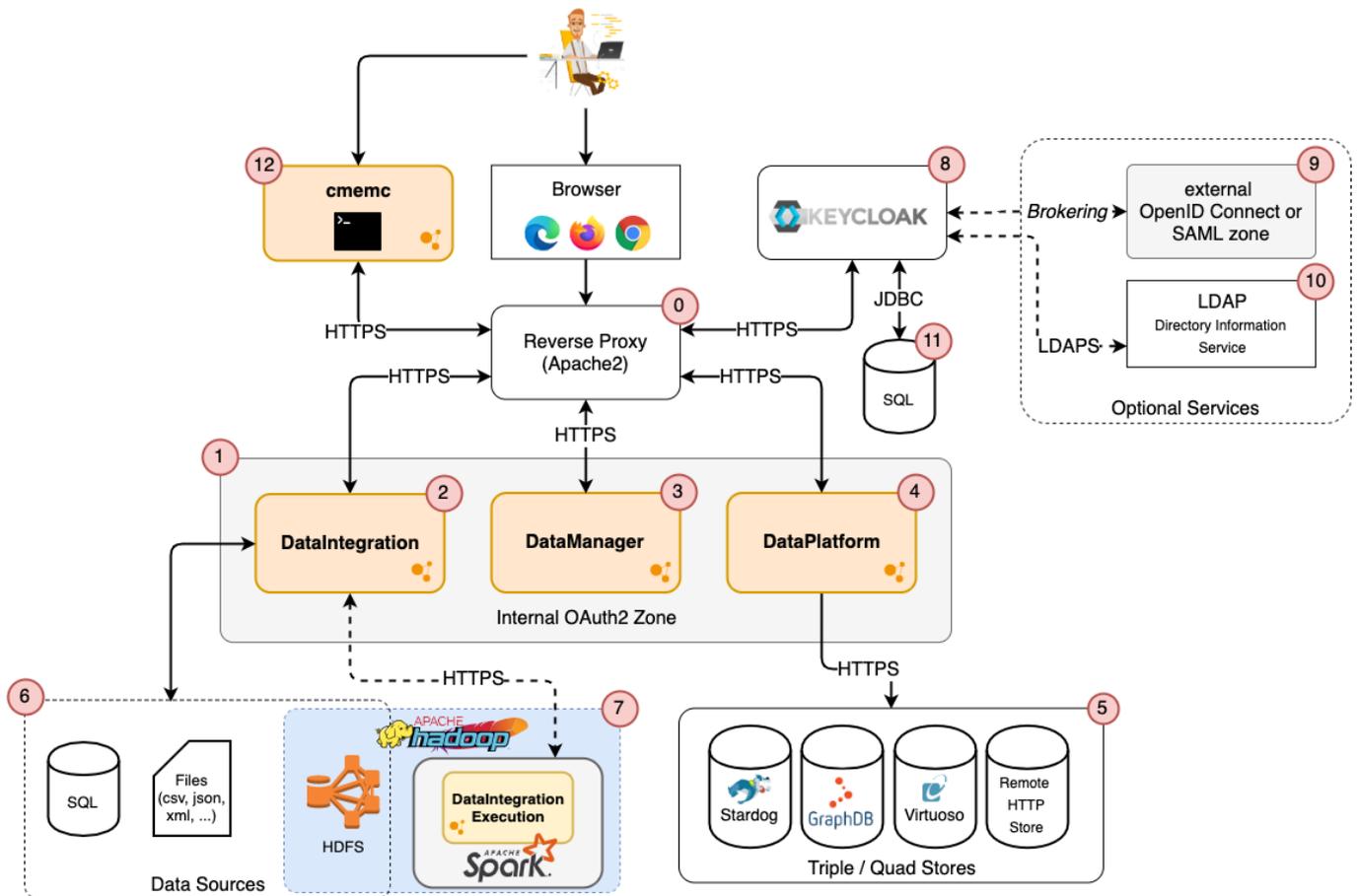
Nachfolgend die Anmelde-Maske des Lifting-Portals<sup>1</sup>, welche mit der Projekt-Infrastruktur insofern integriert wurde, als dass alle Projektmitarbeiter mit Ihrem zentralen Projekt-Account automatisch zu Nutzern des Lifting-Portals wurden.



Im Folgenden werden die einzelnen Komponenten des Lifting-Portals näher beschrieben. Innerhalb des Portals sind diese Komponenten durch einzelne Reiter unterscheidbar. Jede Komponente hat dabei einen klar umrissenen Funktionalitäts-Umfang.

Die technische Architektur des Lifting-Portals stellt sich dabei wie folgt dar:

<sup>1</sup> <https://portal.limbo-project.org>



**DataIntegration** (2) ist die Komponente, welche die Integration mehrerer Datasets in einen Wissensgraph ermöglicht. Datasets in ihrem ursprünglichen Format werden mit RDF-Schemata verknüpft und zu einem Wissensgraphen transformiert. Die Integration erfolgt halb-automatisch basierend auf domänenspezifischen Integrationsregeln und Vokabularen (OWL-Ontologien). Die Komponente unterstützt viele Arten von Datenquellen für die **Integration** (6), z.B. SQL-Datenbanken oder Dateien in verschiedenen Formaten. Diese Dateien können mit der Komponente entweder lokal oder auf einem **Remote-Spark-Cluster** (7) verarbeitet werden.

**DataManager** (3) ist eine JavaScript-Komponente, mit welcher Wissensgraphen erstellt und verwaltet werden können. Die Komponente ist ein generischer Datenbrowser, welcher das Bearbeiten, Erkunden und Abfragen erzeugter Wissensgraphen ermöglicht. DataManager bietet Optionen zum Erstellen gewünschter Datenansichten mithilfe der Shapes Constraint Language (SHACL).

**DataPlatform** (4) ist eine semantische Middleware, welche einen einheitlichen Zugriff auf semantische Wissensgraphen bietet. Darüber hinaus verwaltet DataPlatform die Autorisierung der Benutzer gemäß der im Triple Store definierten Zugriffssteuerungslisten. Die Wissensgraphen werden letztendlich in einem definierten **Triple Store** (5) materialisiert.

**Keycloak** (8) bietet eine offene Authentifizierungsschicht für komplexe Anwendungen. Keycloak kann als Authentifizierungsbroker für bereits vorhandene externe **OpenId-, Connect- oder SAML-Infrastrukturen** fungieren (9). Darüber hinaus unterstützt Keycloak eine Vielzahl von Konfigurationsszenarien für die interne Benutzerverwaltung sowie die Option einer Verbindung zu einem externen LDAP-Server zur **Benutzer- und Gruppensynchronisierung** (10). Keycloak verwendet standardmäßig die eingebettete Java-basierte relationale Datenbank H2 zum Speichern der Konfigurationsdaten. Es wird jedoch dringend empfohlen, stattdessen eine **relationale Datenbank** (11) zu verwenden.

**cmemc** (12) ist ein Open Source Kommandozeilen Werkzeug und richtet sich an Systemadministratoren und Linked Data Experten, welche Aktivitäten auf Wissensgraphen automatisieren und fernsteuern.

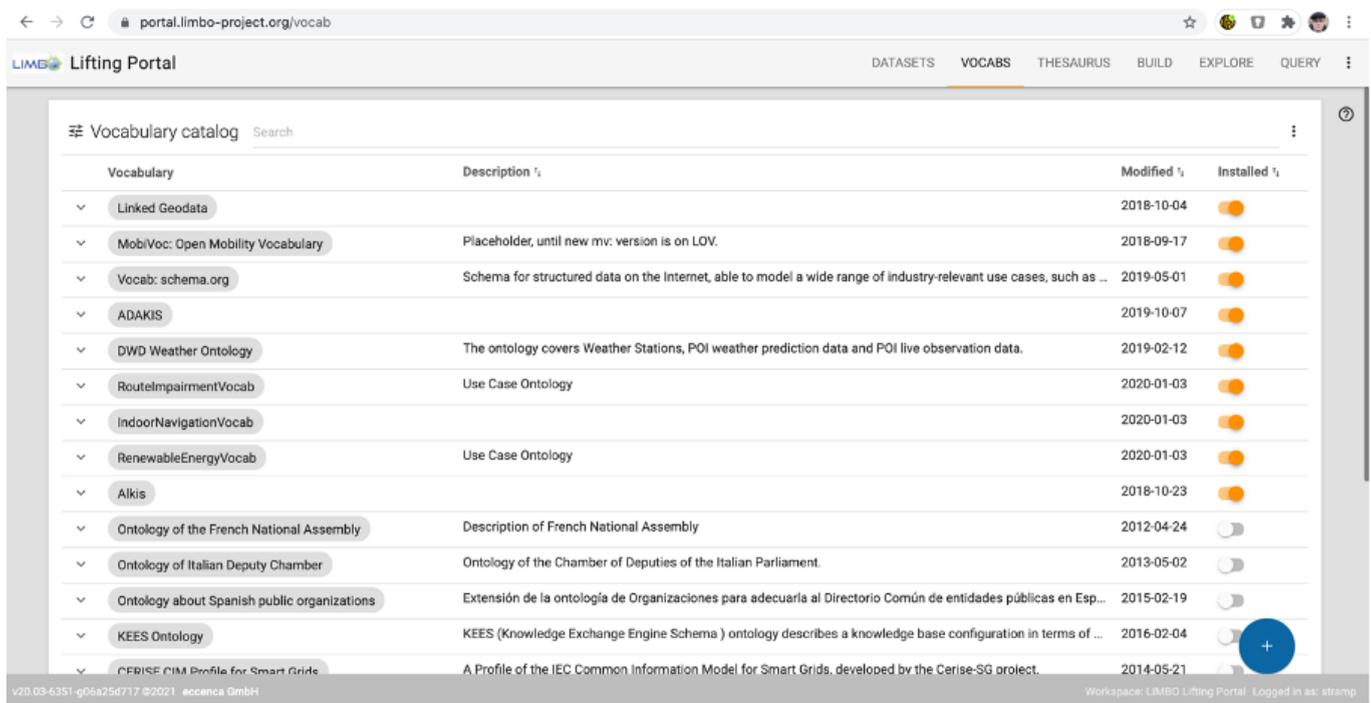
Im Folgenden werden wichtige Funktionsblöcke des Lifting-Portals dargestellt.

### Dataset und Vokabular Management

Ziel dieser Komponenten ist das Management von Vokabularen und Datasets im Portal.

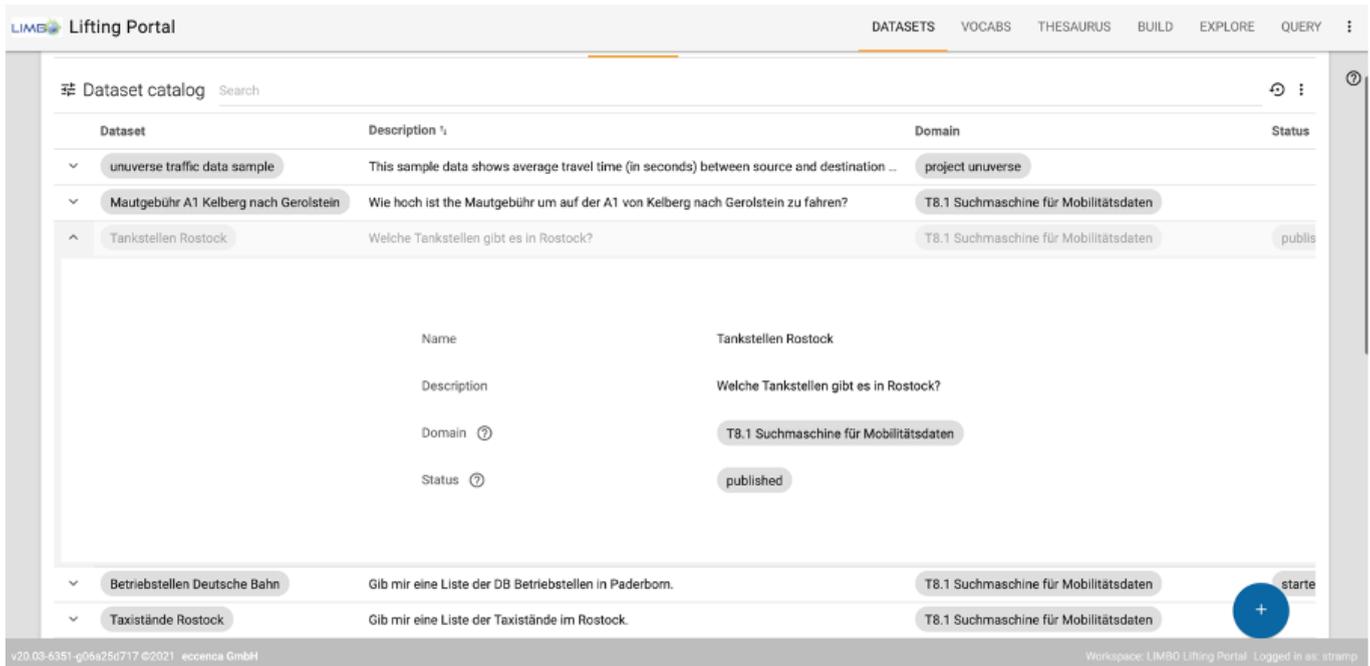
- Als Vokabular werden dabei OWL Ontologien und RDF Schemata verstanden, welche Klassen und Properties semantisch beschreiben. Vokabulare werden im Lifting-Portal als Named Graphs gespeichert. Im LIMBO Prozess kommen sie z.B. aus Vocol oder werden manuell importiert.
- Als Datasets werden dabei Rohdaten bezeichnet, welche im Lifting-Portal zu Wissensgraphen transformiert werden sollen. Datasets können neben Datenbank-Verbindungen (JDBC) insbesondere Datenbank-Dumps oder API-Dumps sein, welche in verschiedenen Formaten vorliegen.

Der Vokabular-Katalog des Lifting-Portals wird im nächsten Screenshot dargestellt.

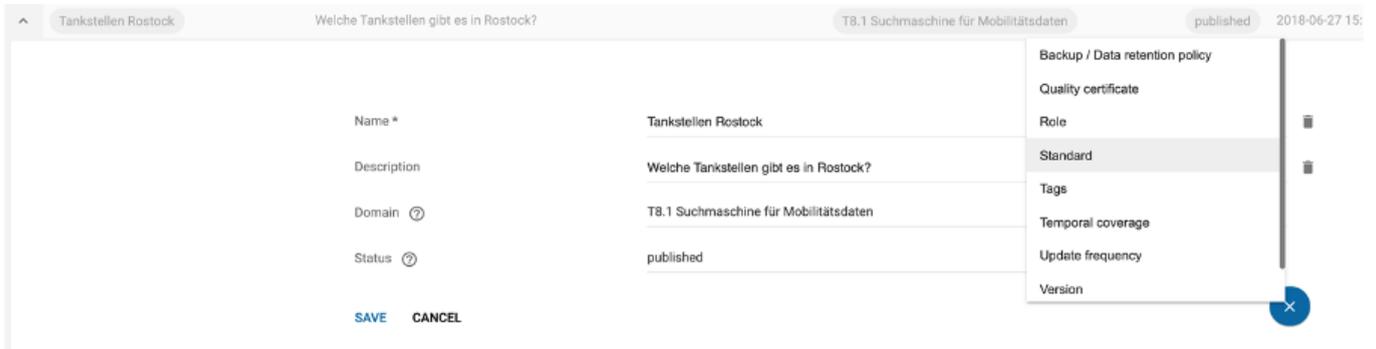


Im Vokabular-Katalog werden Katalog-Einträge zu aus dem Web installierbaren Vokabularen gemangt. Nicht jedes Vokabular muss daher installiert sein. Darüber hinaus können Vokabulare auch manuell durch Hochladen einer RDF-Datei im Turtle-Format (ttl) in den Katalog aufgenommen werden. RDF-Vokabulare sind semantisch strukturiert, so dass Metadaten größtenteils automatisch aus den hochgeladenen Daten extrahiert werden können.

Beim Datasetkatalog werden im Gegensatz dazu Webseiten mit Download-Link resp. Datenbank-Export-Prozesse als Quelle verwendet. Die Metadaten zu diesen Rohdaten werden nicht zwingend mitgeliefert. Aus diesem Grund ist es zwingend erforderlich, reichhaltige Meta-Daten-Benutzerschnittstellen für das Management der Dataset-Metadaten bereitzustellen. Der nächste Screenshot zeigt einen Ausschnitt aus dem aktuellen Dataset-Katalog des Lifting-Portals.



Dabei ist zu beachten, dass der Großteil der Möglichkeiten des angepassten Metadaten-Modells erst im Editier-Modus sichtbar ist. Für das Beispiel des Tankstellen-Datensatzes aus Rostock sieht der detaillierte Katalog-Eintrag wie folgt aus:

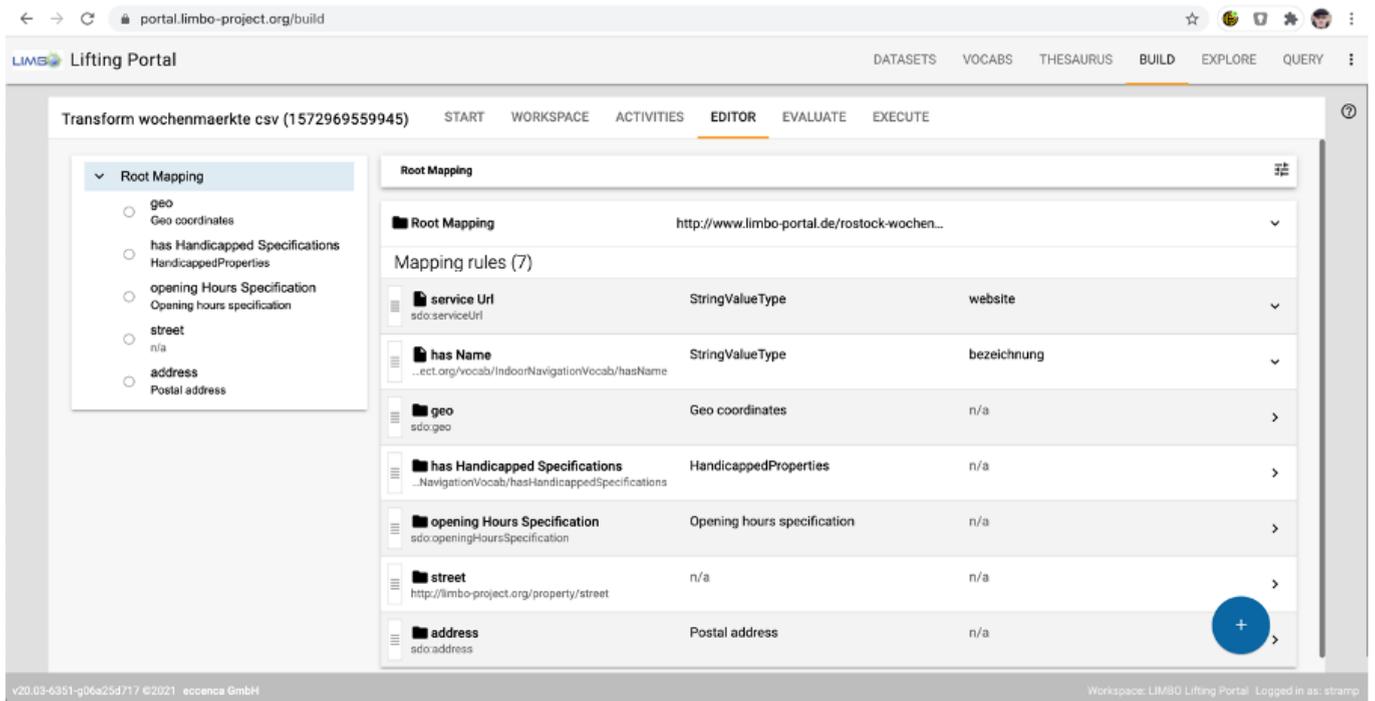


Das gesamte Meta-Daten Modell umfasst dabei 26 Typen und 134 mögliche Eigenschaften. Im späteren Verlauf des Projektes wurden neben den manuell registrierten Datasets auch über 1600 Datasets aus der mCLOUD registriert, indem die Meta-Daten dieser Datasets im Portal hochgeladen wurden.

### Lifting und Linking

Dieser Aufgabenkomplex wird im Lifting-Portal durch mehrere Editor-Interfaces abgebildet. Ziel des Prozesses ist die Transformation von strukturierten, meist tabellarischen oder baumartigen Daten in Wissensgraphen.

Im einfachsten Fall werden dabei Daten-Pfade (z.B. eine konkrete Tabellen-Spalte) Vokabular-Elementen aus einer Ontologie zugeordnet (*Direct Mapping*). Sehr häufig müssen aber mehrere Quell-Pfade gemeinsam zu einer Ziel-Property oder zu einem Ziel-Objekt verarbeitet werden, daher können auch komplexe Transformationen mit Verarbeitungslogik erzeugt werden, welche Microworkflows darstellen. Der nächste Screenshot zeigt das Editor-Interface, welches die Transformation von Rostocker Wochenmärkten beschreibt.

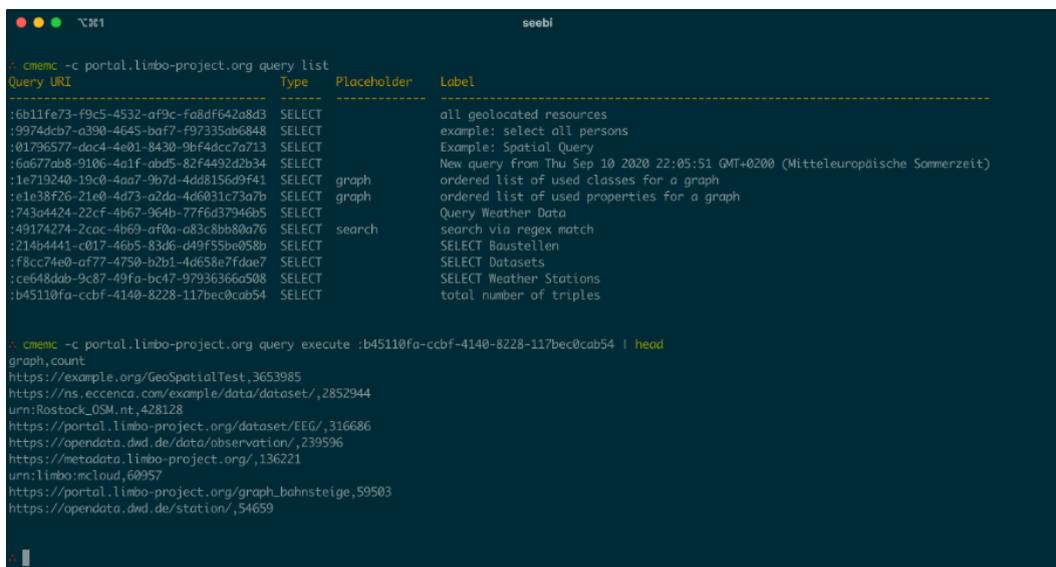


### Integration und Workflows

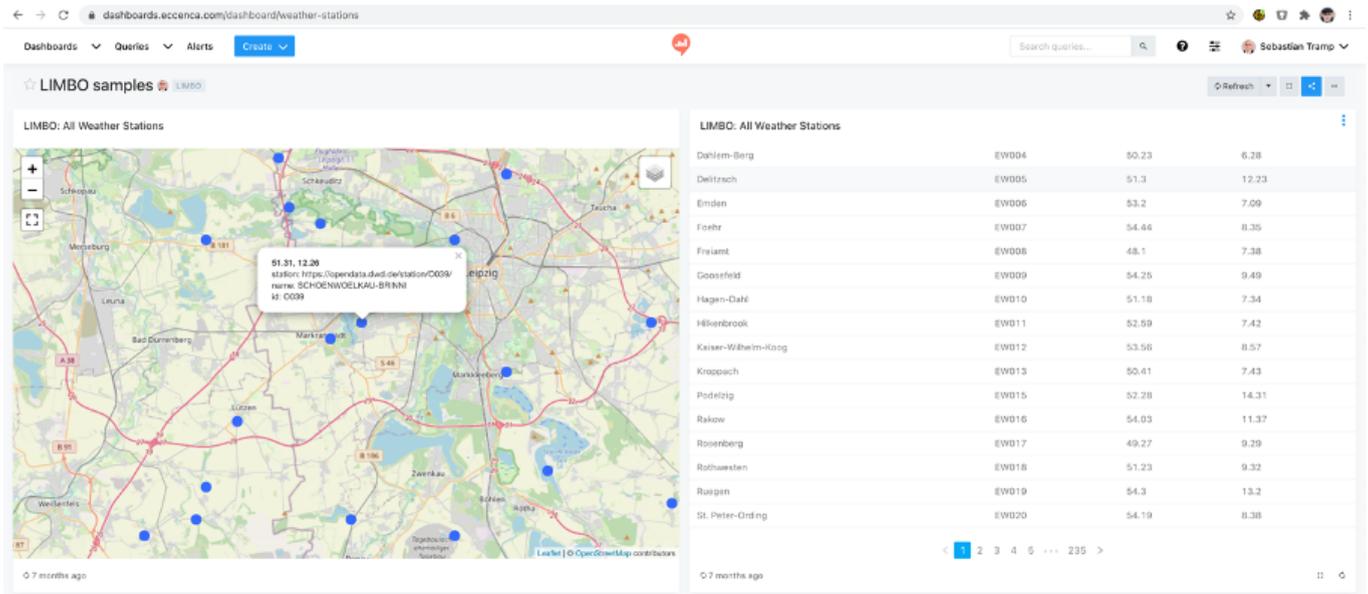
Dieser Aufgabenkomplex umfasst sowohl die Organisation von mehreren Transformationen und Linking-Tasks zu Workflows als auch die Integration des Lifting-Portals in externe Anwendungen. Für die Steuerung und Automatisierung der meisten Prozesse wurde ein offenes Kommandozeilen-Werkzeug entwickelt, welche bereits in der LIMBO-Pipeline verwendet wird. Zu den wichtigsten Funktionalitäten des Werkzeugs zählen:

- Import und Export und Management von Graphen und Vokabularen
- Import und Export und Management von Roh-Daten Datasets
- Start und Verfolgung von Workflows
- Ausführung von SPARQL Anfragen

Nachfolgender Screenshot zeigt dieses Werkzeug, wie es live auf dem Lifting-Portal Daten anzeigt und abfragt.

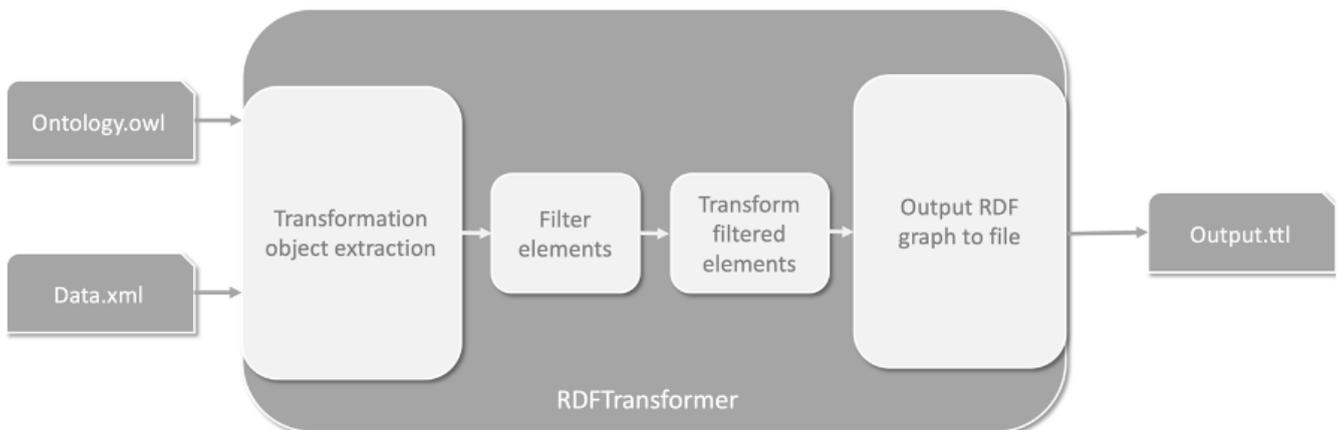


Für die Darstellung und Analyse von vernetzten Daten wurde eine Integration mit dem offenen Dashboard-Werkzeug Redash<sup>2</sup> entwickelt. Dadurch können die Ergebnisse von SPARQL Anfragen an das Lifting-Portal mit komplexen Visualisierungen verknüpft werden. Das Lifting-Portal stellt dafür die Ausführungsumgebung der Anfragen bereit und übermittle die Ergebnisse an das Dashboard-Werkzeug. Das Dashboard-Werkzeug interpretiert die Daten entsprechend der Visualisierungs-Konfiguration, erzeugt Visualisierungen und kombiniert diese zu Dashboards. Anfragen können darüber hinaus auch parametrisiert werden, um gewisse Interaktionsmöglichkeiten zu ermöglichen. Der nächste Screenshot stellt ein auf Redash basierendes Limbo-Dashboard dar, welches Daten basierend auf einem Wissensgraphen über Wetterstationen visualisiert.



### RDFTranslator

Da die zu verarbeitbaren, heterogenen Quelldaten oft sehr groß sind und der Prozess der Umwandlung aufwändig sein kann, ist ein Werkzeug entworfen worden, um den Prozess zu automatisieren. Der RDFTransformer (siehe Abbildung unten) wurde als Python-Bibliothek entwickelt.



Beim Import der Ontologien werden wichtige Informationen zu den jeweiligen Klassen und Eigenschaften extrahiert wodurch das Framework nach entsprechender Parametrisierung dabei unterstützt, XML-Daten unterschiedlicher Quellen in RDF-basierte Graphen umzuwandeln. Diese

<sup>2</sup> <http://redash.io>

Graphen können anschließend über das oben beschriebene Lifting-Portal weiter verarbeitet bzw. über die unten beschriebene Quality Assurance Suite weiter verbessert werden.

## Quality Assurance Suite - QuAC

Das QuAC prüft strukturierte Datensätze wie z.B. SQL, SPARQL, CITRA, (geo)JSON und CSV. Im QuAC bereits vordefinierte, enthaltene Prüfungen und Reports können angepasst und um eigene ergänzt werden. Dies ist sowohl per XML, wie auch über eine einfach zu bedienende GUI möglich. Prüfungen können zusätzlich miteinander verknüpft werden, um auch komplexe Prüfregelein übersichtlich darzustellen und zu steuern. Somit werden Fachanwender wie auch ungeübte Nutzer in die Lage versetzt, effektive Prüfungen und Reports zu erstellen und auszuwerten.

Das QuAC kann als regelmäßiger Prozess, z.B. nächtlich oder als Import-Postprozess, gestartet werden. Verfügbar ist das QuAC als eigenständige Applikation, es kann aber auch in andere Anwendungen wie z.B. eigene GUIs, Java-Beans, Services oder in Datenbanken integriert werden.

Hierzu wurde eine Schnittstelle in Form eines Web-Service realisiert, der die Funktionsaufrufe des QuAC anbietet. Eine Qualitätssicherung kann somit auch innerhalb von Prozessketten ausgeführt werden. Dazu muss eine Konfiguration und die zu prüfende Datenquelle an den Web-Service übergeben werden. Auf diese Art und Weise wurde das QuAC in die Limbo-Prozesskette integriert.

Prinzipiell ist eine Integration an zwei Stellen denkbar:

- Die Daten können vor dem „Lifting“ in ihrem ursprünglichen Datenformat geprüft werden. Dies erfordert jedoch Detailwissen über die Daten und eine Konfiguration der Prüfung muss individuell angefertigt werden.
- Die Daten können nach dem „Lifting“ im RDF-Format geprüft werden.

Im Rahmen des Projektes wurde die zweite Variante in die Limbo-Prozesskette integriert. Dadurch, dass die „gelifteten“/annotierten Daten in ihren Eigenschaften klar definiert sind, können Prüfungen für deren Eigenschaften standardisiert konfiguriert werden.

Für die Limbo-Prozesskette ist es somit möglich, ein Satz an Standard-Prüfungen zu definieren, die nach dem Lifting automatisch angewendet werden. Für diesen Schritt wurde eine speziell angepasste Version des QuAC-Web-Services angefertigt, der als Parameter nur noch eine URL auf den Limbo-Datensatz benötigt und diesen dann auf die Limbo-Regeln prüft.

## Architektur

Kern des Prüfungsframeworks sind die beiden abstrakten Klassen „Metric“ und „Reporter“. Durch die Implementierung dieser Klassen können neue Metriken und Reporter definiert werden. Darauf aufbauend regelt das Framework das Einlesen und Verarbeiten der Konfiguration, den Datenfluss der zu prüfenden Daten und das Zusammenspiel zwischen Metric und Reporter.

Um eine neue Prüf-Metrik zu erstellen, muss die abstrakte Klasse „Metric“ implementiert werden. Dazu gilt es im Wesentlichen zwei Funktionen zu überschreiben: Eine Funktion muss (wenn nötig) den Umgang mit Metrik-spezifischen Konfigurations-Parametern regeln. Die Andere definiert die Prüfung auf einem Element (Row+Column) des Datensatzes.

Der zweite Teil des Frameworks ist die abstrakte Klasse „Reporter“. Sobald eine Metrik ein Element als fehlerhaft markiert hat, wird anschließend der Reporter aufgerufen, der auf den Fehler implementierungsabhängig reagiert. Analog zur abstrakten Klasse „Metric“ muss auch die Klasse „Reporter“ implementiert werden. Das heißt individuelle Argumente, die aus der Konfiguration kommen, müssen behandelt werden und die Aktion auf eine fehlerhafte „row“ muss implementiert werden.

Sowohl Metrik als auch Reporter erfüllen das decorator Pattern. Dadurch kann man sie verschachteln. Das optimiert den Datenfluss indem gleich mehrere Metriken hintereinander auf ein Element des

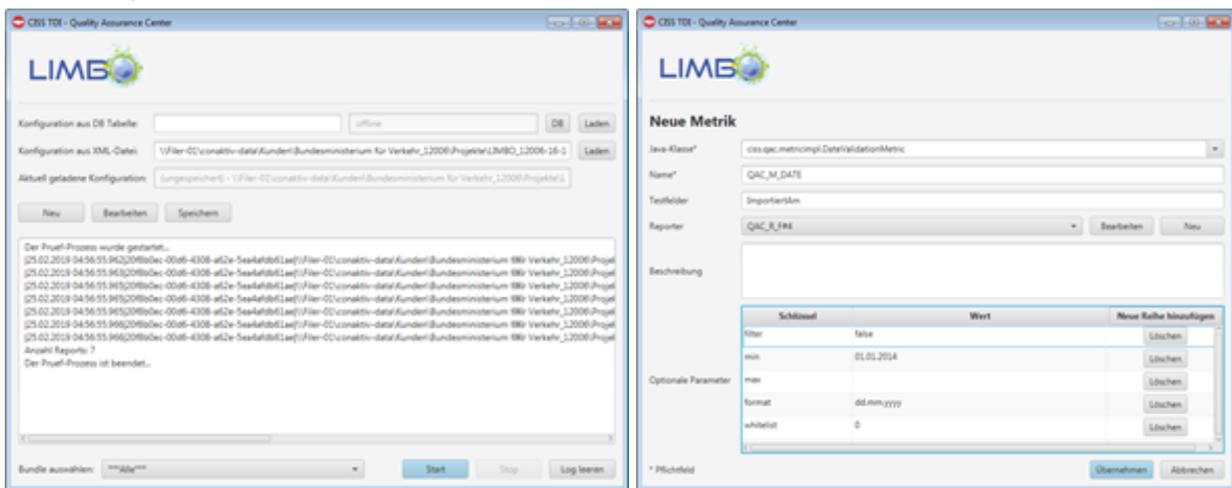
Datensatzes angewendet werden können. Dabei kann auch gefiltert werden, so dass fehlerhafte „Rows“ nicht mehr in der nächsten Metrik behandelt werden müssen.

Für geometrische Prüfungen wurde die externe Bibliothek JTS (JTS Topology Suite, [https://projects.eclipse.org/projects/locationtech.jts]) verwendet. Geometrien können aus WKT und CITRA eingelesen und nach JTS transformiert werden. Anschließend sind Topologische abfragen möglich. Wie z.B.: contains, covers, intersects, touches, equals, usw.

### Konfiguration der Java-Benutzeroberfläche

Alle Einstellungen des QuACs werden über XML konfiguriert. Das beinhaltet die verwendeten Metriken und Reporter, die Daten-Herkunft-Query, die zu prüfenden Attribut-Felder und alle weiteren Metrik-/Reporter-spezifischen Einstellungen. Diese XML-Konfiguration kann auch aus SQL-Tabellen geladen werden. Das ermöglicht die alleinstehende Ausführung innerhalb einer SQL-Datenbank als „java stored procedure“.

Des Weiteren wurde eine Referenz-GUI entwickelt (in JavaFX), mit der zum einen eine bestehende Konfiguration geladen und ausgeführt werden kann (untere Abbildung, linkes Bild), zum anderen Konfigurationen editiert und neu erstellt werden können (untere Abbildung, rechtes Bild). Die Referenz-GUI kapselt die XML-Erzeugung und bildet einen Großteil der Konfigurations-Möglichkeiten des QuACs ab. Eine prototypische API zum Interagieren mit der Konfiguration ermöglicht die Integration des QuAC in eigene GUIs.



Linkes Bild: Start-Maske zum Laden/Erstellen/Ausführen. Rechtes Bild: Maske zum Definieren/Editieren von Metriken..

### Reporter und Datenmodell für Qualitätssicherung

Über die Implementierung der abstrakten Reporter-Klasse kann die Ausgabe der Ergebnisse vielfältig vorgenommen werden. Einige grundlegende Reporter wie. z.B. ein Report auf Terminal/stdout, Report in eine SQL-Tabelle und Report in Log-Dateien sind bereits implementiert und weitere z.B. In GUI-Fenster, als HTML, als Service, usw. sind definiert/prototypisch umgesetzt: Z.B.:

Über die Implementierung des Reporters kann die Ausgabe in ein bestimmtes Datenmodell zur Qualitätssicherung umgesetzt werden. Ein VocabDQV-Reporter wurde prototypisch implementiert.

**Datenkorrektur:** Bisher wurden keine konkreten Korrektur-Algorithmen definiert, da das QuAC zunächst als reines Prüfungsframework gedacht ist. Metrik und Reporter sind jedoch flexibel genug konzipiert, um eigene Metriken und Reporter zu entwickeln, die Reparaturen durchführen können. Dazu müsste eine Metrik zusätzlich zur Prüfung eine Korrektur erarbeiten und der Reporter (oder ebenfalls die Metrik) zusätzlich oder alternativ zum Report die Korrektur in den Datenbestand übernehmen. Der Mechanismus wurde (insbesondere für CITRA) bereits erfolgreich getestet. Auf Basis von SQL-Daten gibt es herstellerspezifische Unterschiede, für die individuelle Anpassungen vorgenommen werden

müssen.

**Prüfalgorithmen:** Folgende Prüfalgorithmen sind u.a. vordefiniert:

- **Attributiv:**
  - Datum: 1) Das Datum muss innerhalb der Spanne „min“ und „max“ liegen, 2) Werte aus einer Whitelist (z.B. „0“, null, „unbekannt“, etc.) sollen erlaubt sein.3) Formatprüfung: Das Datum muss einem Format-String entsprechen.
  - Lookup: 1) Ergebnis ist OK wenn Wert in der Liste vorkommt. 2) Invers: Ergebnis ist nicht OK wenn Wert in Liste vorkommt.
  - Nummer-Validierung: 1) Nummer entspricht einem bestimmten Datentyp, 2) Dezimal-/Tausender-Trennzeichen, 3) Anzahl der Nachkommastellen, 4) Exponential-Schreibweise
  - Regexp: 1) Ergebnis ist OK wenn Wert dem Regexp genügt, 2) Invers: Ergebnis ist nicht OK wenn Wert dem Regexp genügt.
  - Vollständigkeits-Prüfung: 1) Ein Wert einer Liste muss mindestens einmal im Result-Set vorkommen.
  - Einzigartigkeits-Prüfung: 1) Kommt ein Wert mehr als einmal im Datensatz vor?
- **geometrisch:**
  - Diverse Topologie-Prüfungen, z.B.: 1) Liegt ein Objekt auf einer Trasse/Leitung? 2) Trennen Reduzierstücke die Leitungen auf? 3) Existieren Punkte mit identischen Koordinaten (Doppelerfassungen)?
  - Umring-Prüfung: 1) Liegt die Geometrie innerhalb einer Fläche (Umring)?
- **attributiv und geometrisch kombiniert:**
  - Diverse Topologie-Prüfungen, z.B.: 1) Leitung mit Attribut „Hausanschluss“ ist verbunden mit Hausanschluss und Abzweig-Muffe.

Für die zukünftige Anwendung auf mFUND Daten und/oder neu erhobenen LIMBO-Datensätzen wurde eine Demo-Konfiguration definiert, die sowohl Sachdaten als auch Geometrien prüft. Diese Demo prüft, ob sich eine Koordinate innerhalb von Deutschland befindet und innerhalb einer gewählten Datums-Spanne liegt. Als Datenbasis könnte dazu eine SPARQL-Query dienen. Die folgenden Abfragen veranschaulichen den Nutzen einer solchen Prüfung. Um das Beispiel kurz zu halten wurde auf eine komplexe anwendungsspezifische Abfrage an dieser Stelle verzichtet

Beispiel SPARQL-Query:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?url (sample(str(?gn)) as ?givenname) (sample(str(?sn)) as ?surname)
(sample(str(?bd)) as ?birthday) (sample(?p) as ?point)
WHERE {
?url dbo:award <http://dbpedia.org/resource/Nobel_Prize_in_Physics> .
?url dbo:birthPlace ?bp .
?url dbo:birthDate ?bd .
?url foaf:givenName ?gn.
?url foaf:surname ?sn.
?bp geo:lat ?lat .
?bp geo:long ?lon .
BIND(CONCAT ("POINT(", str(?lon) , " " , str(?lat),")") AS ?p) .
}
Group by ?url
LIMIT 1000
```

Beispiel Konfiguration (XML):

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<config>
  <metric_definition>
    <metric_entry child_metric_id="0" child_metric_table="QAC_M_UMRING" id="0"
metric_class="ciss.qac.metricimpl.DateValidationMetric" name="QAC_M_DATE" reporter_id="0"
reporter_table="QAC_R_TERMINAL" testcols="birthday">
      <parameter key="min" value="1800-01-01"/>
      <parameter key="max" value="2000-01-01"/>
      <parameter key="format" value="yyyy-dd-mm"/>
    </metric_entry>
    <metric_entry child_metric_id="" child_metric_table="" id="0"
metric_class="ciss.qac.metricimpl.GeometryContainsMetric" name="QAC_M_UMRING"
reporter_id="0" reporter_table="QAC_R_TERMINAL" testcols="point">
      <parameter key="file_path" value="D:\Temp\poligon_ger.txt"/>
      <parameter key="inverse" value="true"/>
      <parameter key="input_type" value="wkt_file"/>
    </metric_entry>
  </metric_definition>
  <reporter_definition>
    <reporter_entry child_reporter_id="" child_reporter_table="" id="0"
name="QAC_R_TERMINAL" reportcols="givenname, surname, birthday, point"
reporter_class="ciss.qac.reporterimpl.TerminalReporter">
      <parameter key="status" value="Info"/>
    </reporter_entry>
  </reporter_definition>
  <config_definition>
    <config_entry bundle_id="1" data_src="[SPARQL-Query siehe oben]" id="0"
metric_id="0" metric_table="QAC_M_DATE"/>
  </config_definition>
</config>

```

## Web-Applikation

Neben der Java-GUI existiert zudem eine Web-Applikation für das QuAC. Die Webapp soll für die Nutzer leichter zugänglich sein als eine dedizierte Java-Anwendung. Außerdem ermöglicht die Webapp auch die Onlinesicherung einzelner Konfigurationen. Bei den aktuellen Offline-Varianten des QuACs müssen das Lesen und Schreiben der Konfiguration vom Nutzer selbst geregelt werden.

Ziel war es, mit der Webapp dem Nutzer alle Funktionen des Programms zur Verfügung zu stellen und gleichzeitig das Programm für unterschiedliche Nutzergruppen bereitzustellen.

Es ist möglich, die Konfiguration der einzelnen Metriken, Reporter und Datenbankanbindungen in einem XML-Format an die Webseite zu übergeben und die Datensätze zu testen. Diese Variante entspricht dem bisherigen Vorgehen und ist besonders an IT-Affinierte Nutzer gerichtet und für diese sehr von Vorteil. Weiterhin können Metriken, Reporter und Datenbankanbindungen auf der Webseite in einer grafischen Oberfläche zusammengestellt werden. Diese Variante ist besonders für Nutzer mit weniger IT-Erfahrung gedacht. Auch schon vorgefertigte Prüfungen können benutzt werden, um Datensätze zu Testen. Nutzer können sich eigene Prüfungen, Metriken, Reporter, Regeln und Datenbankanbindungen jederzeit in Tabellen anzeigen lassen, diese bearbeiten, löschen und als XML-Dateien exportieren. Hier besteht ebenfalls die Möglichkeit, sich neue Prüfungen und Prüfungsdatensätze anzulegen oder vorhandene auf aktiv oder inaktiv zu setzen, um die Übersicht in der grafischen Darstellung zu verbessern. Prüfungen können exportiert und so in anderen Prozessen wiederverwendet werden. Diese Funktion richtet sich an Nutzer, die gemeinsam, mit unterschiedlichen Rollen an der Qualitätssicherung arbeiten. Hierzu ist auch ein Rollenmanagement vorgesehen, welches eine „Manager“-Rolle und eine Mitarbeiter-Rolle unterstützt. Der „Manager“ konfiguriert die Regeln und Prüfungen. Diese Rolle erfordert detailliertes Wissen über die Daten und die Konfigurationsmöglichkeiten des QuACs. Die „Mitarbeiter“-Rolle kann bereits konfigurierte Prüfungen verwenden und auf den Datensatz anwenden. Für diese Rolle ist jedoch nur oberflächliches Wissen über das QuAC erforderlich.

Folgendes ist derzeit möglich (s. unten stehende Abbildungen):

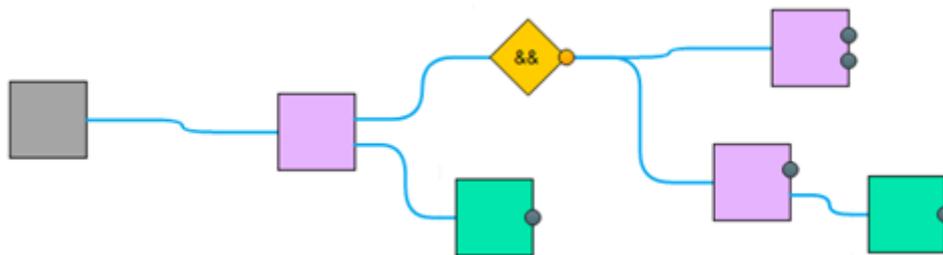
- Es können XML-Dateien entweder per drag-and-drop oder durch Hochladen eingelesen werden. Weiterhin kann der zu prüfende Datensatz hochgeladen werden.
- In der grafischen Oberfläche können Metriken, Reporter und Datenbankanbindungen erstellt,

gespeichert und verändert, sowie komplexe Prüfungen zu konfigurieren und gespeichert werden.

- Bei Tabellen wird eine Anzeige vorhandener Metriken, Reporter, Datensätze und Prüfungen unterstützt. Weiterhin können neue erstellt, verändert, gelöscht, exportiert und deren Sichtbarkeit verändert werden. Die Funktionen Öschung, Export und Änderung der Sichtbarkeit kann mittels Markierung auch auf mehrere Datensätze angewandt werden.
- Es können Datensätze hochgeladen und anschließend mittels einer oder mehrerer Prüfungen analysiert werden..

Auswählen	Aktivieren	ID	Bezeichnung	Bundlename	Bearbeiten	Löschen
<input type="checkbox"/>	<input checked="" type="checkbox"/>	0	Name0	Prüfung0		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	Name1	Prüfung1		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	2	Name2	Prüfung2		

Beispiel der Tabellen in welchen alle Prüfungen/Metriken/Reporter/Regeln/Regel-Ketten und Datenbankverbindungen aufgelistet sind



Beispiel der grafischen Zusammenstellung einer Prüfung. Grau stellt die Datenbankverbindung, lila die Metriken und grün die Reporter dar

### OGC-konforme Bereitstellung von Geodaten durch ein Data Warehouse (GDW)

Für die Erweiterung eines CISS-GDW um Schnittstellen zum Semantischen Web, insbes. die Bereitstellung der Geodaten als Linked Data im RDF-Format, wurden die von der W3C Arbeitsgruppe RDB2RDF<sup>3</sup> erarbeiteten Standards herangezogen. Diese erlauben die Umwandlung vorhandener relationaler Datenbestände in sog. „virtuelle“ RDF-Datenbestände, d.h. ohne dass die Daten dafür vorab umformatiert und damit auch dupliziert werden müssen. Für die Abbildung der relationalen Daten auf das RDF-Format gilt im einfachsten Fall eine Default-Spezifikation („direct mapping“<sup>4</sup>), erheblich leistungsfähiger und daher zu bevorzugen sind aber anwendungsspezifische Abbildungen („customized mapping“), die in der RDF-basierten Sprache R2RML<sup>5</sup> konfiguriert werden.

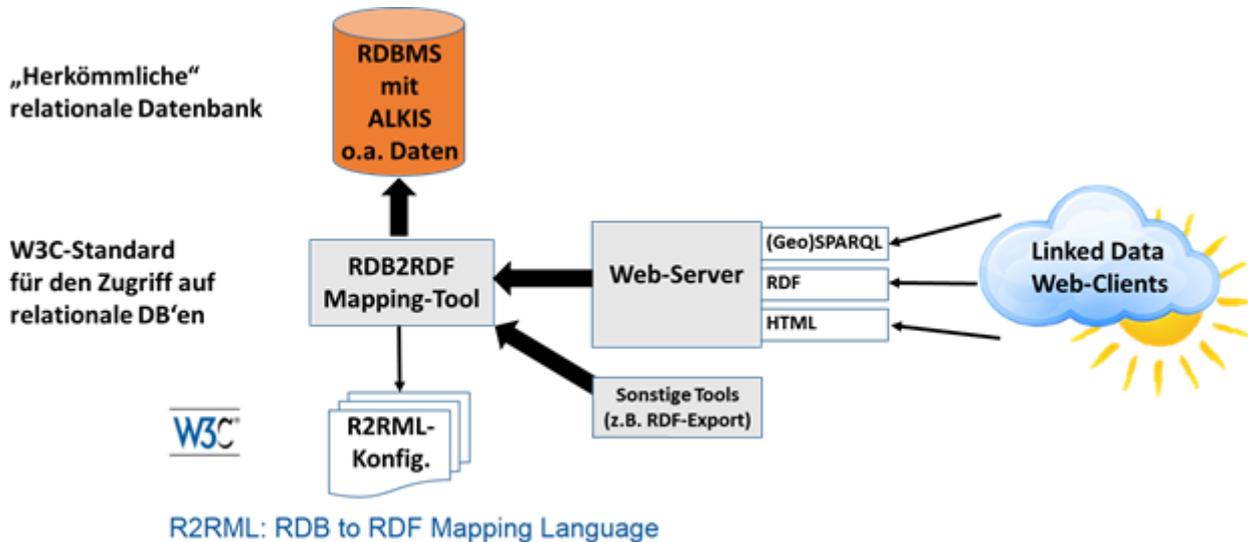
Als Vorteil dieser Vorgehensweise wird vor allem erwartet, dass bestehende GDWs evolutionär weiterentwickelt werden können. Damit ist gemeint, dass vorhandene Datenbestände ebenso wie die dazugehörigen Konverter und Schnittstellen weiter verwendbar und alle Services und Workflows

<sup>3</sup> Siehe <http://www.w3.org/2001/sw/rdb2rdf/>

<sup>4</sup> Siehe <http://www.w3.org/TR/rdb-direct-mapping/>

<sup>5</sup> Siehe <http://www.w3.org/TR/r2rml/>

erhalten bleiben sollen. Neu zu entwickeln sind dann im Wesentlichen die R2RML-Konfigurationen, wozu jedoch nicht nur das Erstellen der Konfigurationsdateien, sondern auch die Definition der darin verwendeten Vokabulare bzw. Ontologien gehört. Das Konzept wird durch die nachfolgende Abbildung verdeutlicht:



## Datenhaltung

Für die Datenbank, als zentrale Komponente eines GDW, wird Oracle eingesetzt. Dies hat sich bei CISS seit vielen Jahren in zahlreichen Projekten bewährt und so können auch bestehende Lösungen evolutionär zu einem GDW mit Linked Data erweitert werden (linked open geodata warehouse, LOGDW). Die zusätzlichen Komponenten zur Speicherung, Abfrage und für den Export von RDF-Daten werden mit Oracle Spatial and Graph und dem Fuseki-Server aus dem Apache Jena Framework realisiert.

Vor allem im Umfeld heterogener Systemlandschaften kann jedoch die Entscheidung für nur einen Datenbankhersteller auch Nachteile mit sich bringen. Um die Möglichkeiten der Datenhaltung noch flexibler gestalten zu können, wurde untersucht, ob auch eine Kombination verschiedener Datenbanktechnologien möglich ist. Außer Oracle kommt dafür vor allem die OpenSource-Datenbank PostgreSQL in Betracht, für die zahlreiche Anwendungen und Tools – i.d.R. ebenfalls als OpenSource – im Markt verfügbar sind. Daher wurden Technologien für den wechselseitigen Zugriff zwischen PostgreSQL und Oracle Datenbanken untersucht. Es stellte sich heraus, dass dies in beide Richtungen möglich ist: Es kann sowohl ein Zugriff von PostgreSQL auf Oracle als auch umgekehrt ein Zugriff von Oracle auf PostgreSQL erfolgen.

**PostgreSQL Foreign Data Wrapper for Oracle:** PostgreSQL bietet in Form der sog. „Foreign Data Wrapper (FDW)“ Funktionalitäten für den Zugriff auf externe Datenbanken an, die im Teil „SQL/MED (Management of External Data)“ des SQL-Standards definiert sind<sup>6</sup>. Normalerweise läuft die externe Datenbank dabei auf einem anderen Server als PostgreSQL (remote server). Über PostgreSQL Extensions (Plugins) werden diverse Typen von externen Datenbanken unterstützt; die Seite [https://wiki.postgresql.org/wiki/Foreign\\_data\\_wrappers](https://wiki.postgresql.org/wiki/Foreign_data_wrappers) führt z.Zt. weit über 50 verschiedene Datenbanktypen auf.

Die PostgreSQL Extension für den Zugriff auf Oracle heißt „Oracle Foreign Data Wrapper“ (Oracle

<sup>6</sup> <https://en.wikipedia.org/wiki/SQL/MED>

FDW<sup>7</sup>). Sie ist sehr einfach zu installieren und erlaubt u.a. auch den Austausch von Geometrien, z.B. über das ISO-standardisierte Well-Known-Text Format (WKT<sup>8</sup>).

**Oracle Database Gateway for ODBC:** Einen Zugriff in umgekehrter Richtung von Oracle auf PostgreSQL ermöglicht das „Oracle Database Gateway for ODBC“<sup>9</sup>. Wie der Name schon andeutet, realisiert dieses Gateway nicht speziell einen direkten Zugriff auf eine PostgreSQL-Datenbank, sondern es greift auf eine entsprechend dem ODBC-Protokoll bereitgestellte Schnittstelle zu. Über das Oracle Database Gateway for ODBC ist deshalb nicht nur ein Zugriff auf PostgreSQL möglich, sondern auf alle Datenquellen, für die ein geeigneter ODBC-Treiber verfügbar ist. Derartige Treiber werden aber nicht von Oracle, sondern von der Fremdsoftware zur Verfügung gestellt, hier also von PostgreSQL.

Bisherige Erfahrungen und Bewertung: Da das Oracle Database Gateway for ODBC eine komplexere Installation und Konfiguration erfordert, wird der Zugriff von PostgreSQL auf Oracle mit dem Oracle FDW bevorzugt. Durch diese Kombination der beiden Datenbanken kann das Beste aus beiden Welten für das Linked Open Geodata Warehouse eingesetzt werden.

### **Bereitstellung von ALKIS-Daten und von COSMO-REA-Daten über einen SPARQL-Endpunkt**

Mit dem bereits als CISS-Produkt vorhandenen ALKIS-Konverter für Oracle wurde ein ALKIS-Datenbestand in einem relationalen Datenmodell eingerichtet, der als Testdatenbestand für die RDB2RDF-Tools dient. Dafür wurden zunächst nur das Direct Mapping sowie sehr einfache Beispiele von Customized Mapping Konfigurationen genutzt. Die exemplarische Bereitstellung formalisierter ALKIS-Daten über eine entsprechende Ontologie, wurde auf Basis erster Daten durchgeführt (vgl. Kapitel "Daten und Ontologien").

Für die Datenhaltung der Reanalysedaten fiel die Entscheidung zu Gunsten von PostgreSQL mit PostGIS-Erweiterung. Die Daten werden im Raster-Format (wahlweise aus GRIB, netCDF oder GeoTIFF) vorverarbeitet und in die Datenbank importiert. Die für die Abfragen und den Zugriff durch Dienste benötigte vektorisierte Form wird erst innerhalb der Datenbank realisiert.

### **Bereitstellung von Geodiensten**

Zur Implementierung als Webservice standen zwei eingeführte Technologien zur Auswahl, die für die Anforderungen geeignet erschienen: Sowohl MapServer<sup>10</sup> als auch GeoServer<sup>11</sup> sind freie Mapserver-Projekte, die Geodienste u.a. gemäß der Spezifikationen des Open Geospatial Consortium (OGC) zur Verfügung stellen. Darüber hinaus können diese individuell erweitert werden. Da zum Zeitpunkt der ersten Evaluierungen die endgültige Entscheidung für eine bestimmte Datenbanktechnologie (ORACLE/PostgreSQL) als Datenquelle noch nicht gefallen war, sollten beide Technologien auch beide Datenbank-Systeme anbinden können.

In prototypischen Implementierungen und Tests konnten beide Mapserver-Technologien diese Anforderung erfüllen.

Die im Anwendungsszenario verwendeten COSMO-ReAnalyzedaten (und Wetterdaten allgemein) werden häufig in den Formaten GRIB / netCDF verbreitet. Da der GeoServer mittels Erweiterungen (sog. Extensions) diese Daten auch direkt einbinden kann und darüber hinaus zukünftig zu erwarten ist, dass dieser sich im Meteorologischen Umfeld weiter etablieren wird, fiel die Entscheidung zu Gunsten der GeoServer-Technologie.

---

<sup>7</sup> [https://laurenz.github.io/oracle\\_fdw/](https://laurenz.github.io/oracle_fdw/)

<sup>8</sup> [https://en.wikipedia.org/wiki/Well-known\\_text\\_representation\\_of\\_geometry](https://en.wikipedia.org/wiki/Well-known_text_representation_of_geometry)

<sup>9</sup> <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/odbcu/index.html>

<sup>10</sup> <https://mapserver.org>

<sup>11</sup> <https://geoserver.org>

Vorbereitung der COSMO-Reanalyse Daten: Die Bereitstellung von Daten (und Metadaten) erfolgte bevorzugt in der Datenbank. Dazu wurde eine Prozesskette definiert und implementiert, um die Daten der COSMO-Reanalyse in eine Datenbank zu importieren und die Daten anschließend als Dienst zur Verfügung zu stellen.

Die Konvertierung erfolgte unabhängig von der verwendeten Datenbanktechnologie und wurde prototypisch für eine PostgreSQL/PostGIS-Datenbank (Versionen 9.6 und 10) implementiert und ausführlich getestet. Die Konvertierung wurde so aufgebaut, dass sie nahezu unverändert auch für ORACLE übernommen werden kann. Dazu wurden die Daten in ein einfaches CSV-Format überführt und anschließend in die Datenbank importiert.

Durch das European Center for Medium-Range Weather Forecasts (ECMWF) wird das Softwarepaket ecCodes<sup>12</sup> zur Verfügung gestellt, das die Daten der COSMO-ReAnalyse (GRIB-Format) verlustfrei nach CSV konvertiert. Die im GRIB-Format enthaltenen Metadaten können durch ecCodes als ASCII-Textdaten übernommen werden.

Der Import in die PostgreSQL-Datenbank erfolgte anschließend mittels "COPY", für ORACLE steht analog ein Import mittels "SQL-Loader" zur Verfügung.

Als Datenmodell kam das Schneeflocken-Schema zur Anwendung. Wesentliche / notwendige Inhalte der Metadaten (z.B. meteorologischer Parameter, Datum / Zeit) wurden sowohl zur Ablaufsteuerung verwendet, als auch als Daten direkt in die Faktentabelle übernommen. Die eigentlichen meteorologischen Werte wurden in die Dimensionstabelle(n) integriert. Die übergreifenden Daten (3-dimensionale Koordinaten des Gitterkubus) wurden in eigene Koordinatentabellen übernommen und sind für alle Inhalte der Dimensionstabellen gemeinsam gültig.

In seiner Gesamtheit stehen damit alle für den Dienst festgelegten Informationen zur Verfügung:

- der meteorologische Parameter [Wind, Temperatur, ...] - in den Dimensionstabellen
- die Position [als Koordinate] - in der Koordinatentabelle
- der Zeitpunkt [als Datums-/Zeitstempel] - in der Faktentabelle

Die Konvertierung ist in der aktuellen Implementierung noch Zeit-/Rechenaufwendig. Da die Daten der COSMO-Reanalyse weitgehend statisch sind, stellt dies kein Problem dar.

Das Datenmodell wurde auf eine hohe Performanz bei der Datenabfrage optimiert. Diese konnte in ersten Performanz-Tests erfolgreich nachgewiesen werden, bietet trotz alledem noch Potential für eine weitere Verbesserung.

**Bereitstellung eines WMS/WFS:** Prototypisch wurde in der Kombination GeoServer / PostgreSQL-Datenbank ein WMS-Dienst eingerichtet und ausführlich getestet (siehe nachfolgende Abbildung). Darüber hinaus wurden die Möglichkeiten untersucht, mit GeoServer direkt auf die Ausgangsdatenformate GRIB und netCDF zuzugreifen.

Dazu wurde die PostgreSQL-Datenbank als Datenquelle an den GeoServer angebunden, so dass alle mit einem Geodatum (Koordinate, ggf. Datums-/Zeitstempel) verknüpften Daten dem GeoServer zur Verfügung stehen. Basierend auf den durch das Anwendungsszenario definierten Fragestellungen (Verteilung der Windgeschwindigkeit in einem Raumgebiet / Zeitreihe der Windböen an einem Ort) wurden in der Datenbank zur Selektion Views angelegt. Diese wurden als Datenquelle an den Dienst angebunden.

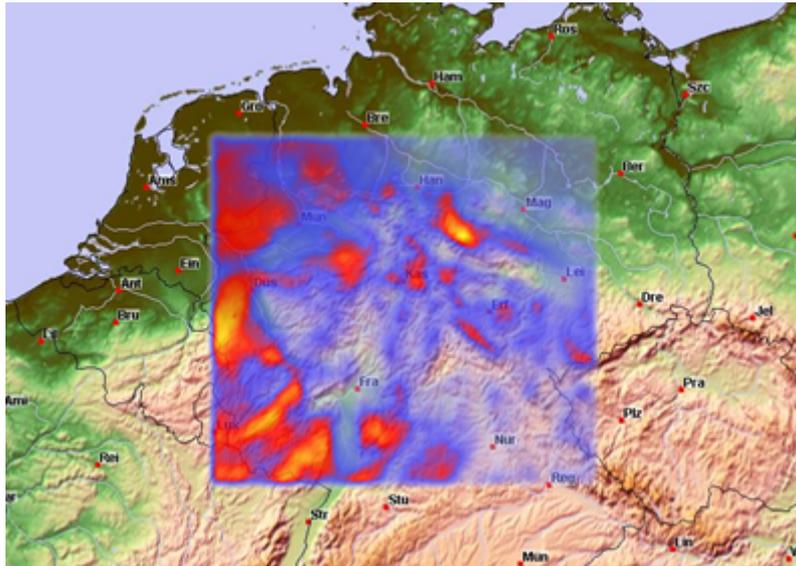
Der Zugriff auf GRIB / netCDF und die Darstellung als WMS-Dienst waren erfolgreich. Priorisiert wird jedoch, eine Datenbank als Datenspeicher des Dienstes zu nutzen. Der WMS Dienst wird jedoch für das LIMBO-Szenario Smart Charging verwertet, welcher von YellowMap definiert wurde.

Basierend auf den durch das Anwendungsszenario definierten Fragestellungen (Verteilung der Windgeschwindigkeit in einem Raumgebiet / Zeitreihe der Windböen an einem Ort) wurden in der Datenbank zur Selektion Views angelegt. Diese wurden als Datenquelle an den Dienst angebunden und

---

<sup>12</sup> <https://confluence.ecmwf.int/display/ECC>

von diesem dargestellt.



COSMO-Reanalysedaten, bereitgestellt als WMS

### Datenmodellierung

Die im Rahmen des Anwendungsszenarios vorliegenden Daten der COSMO-ReAnalyse können nach folgenden Kriterien klassifiziert / selektiert werden:

- Meteorologischer Parameter (z.B. Wind (Richtung, Stärke, Böe) / Temperatur / ...)
- Gültigkeit (Datum / Zeit)
- Ort (Koordinate und Höhe)

Für einen nach diesen Kriterien optimierten Zugriff bietet sich ein Datenmodell in Anlehnung an das „Schneeflocken“-Schema an. Dabei werden um eine zentrale „Faktentabelle“ beliebig viele „Dimensionstabellen“ angeordnet. Die Dimensionstabellen enthalten den Wert des Parameters (z.B. die Windgeschwindigkeit). Über einen Schlüssel wird dieser aus der Faktentabelle referenziert, in der u.a. die Informationen zur Gültigkeit und Ort und ggf. weitere Metadaten gespeichert sind. Zwar ist diese Datenhaltung nicht vollständig normalisiert, jedoch sehr performant bei der Datenabfrage und entspricht damit den gestellten Anforderungen.

Im Rahmen der Implementierung wurde das Datenmodell so erweitert, dass die Koordinaten des 3D-Gitters, auf dem die Daten vorliegen, aus der Faktentabelle in eigene Tabellen ausgelagert wurden. Dadurch wurde das Datenmodell in ein (für ein DWH typisches) „Schneeflocken“-Schema überführt.

Mittels Datenbank-Views wurden (z.Z. noch als statische) Abfragen /Views definiert und implementiert, die einen (OGC-konformen) Zugriff auf die selektierten Daten ermöglichen. Die Abfragen basieren auf den im Rahmen des Anwendungsszenarios definierten typischen Fragestellungen u.a. auf Basis der Interview-Ergebnisse. Es folgen zwei Beispiele:

Fragestellung aus Sicht eines Windkraftanlagen-Projektierers: „Welches mittlere Windpotential gibt sich in einem bestimmten Zeitraum in einem vorgegebenen Raumbereich und in einer bestimmten Höhe?“

Diese Abfrage selektiert in der Datenbank nach den folgenden Parametern und ermittelt entsprechende Isolinien:

- dem Parameter „Windgeschwindigkeit“ (keine Relevanz der Windrichtung)
- einem Raumgebiet (Gebiet definiert durch Koordinatenbereiche)
- einer bestimmten Höhe (exakte Angabe der Höhe)
- einem bestimmten Zeitraum (gegeben durch Start-/Endzeit)

Fragestellung aus Sicht eines Versicherungs-Gutachters: „Gab es an einem vorgegeben Ort zu einem gegeben (kurzen) Zeitraum Windböen, die über einem bestimmten Grenzwert lagen?“

Diese Abfrage selektiert in der Datenbank nach folgenden Parametern und interpoliert ggf. das

Ergebnis:

- dem Parameter „maximale Windgeschwindigkeit / Böe“ (keine Relevanz der Windrichtung), ggf. eingeschränkt auf Werte über einem gegebenen Grenzwert
- an einem exakten Punkt (Ort definiert durch Koordinate)
- in einem bestimmten Zeitraum (gegeben durch Start-/Endzeit)

### Validierung der Teilergebnisse

Auf Basis der durchgeführten Arbeiten konnten sowohl die Mechanismen zum Datenimport, der direkte Zugriff auf die Daten in der Datenbank und der Zugriff durch Dienste getestet werden. Die Ergebnisse und die ggf. daraus folgenden Tätigkeiten werden in der Folge beschrieben.

Der Konverter zur Datenintegration von COSMO-ReAnalysen überführt die Daten vollständig in das definierte und implementierte Datenmodell. Dazu wurden die Daten des 3D-Koordinatengitters komplett, und meteorologische Daten in den Mengen importiert, dass eine Aussage zur Machbarkeit zum Aufbau der Abfragen möglich war.

Die dabei evaluierten Import-Zeiten sind in Bezug auf einen vollständigen Import der aktuell zur Verfügung stehenden Daten unkritisch. Eine Verbesserung der Performanz wird zwar angestrebt, wird aber nicht als sehr dringlich angesehen, da die Daten nahezu statisch / unverändert bleiben.

Die sich an den definierten Fragestellungen orientierenden Abfragen lieferten sehr gute Ergebnisse. Mit einer weiteren Optimierung des Datenmodells und der Hardware sind noch weitere Verbesserungen zu erwarten.

### Stichwortsuche und Question Answering

Für die Stichwortsuche und das Question Answering (QA) wurde das Framework TeBaQA<sup>13</sup> entwickelt, welches auf einer Kombination aus SPARQL-Query Templates und maschinellem Lernen basiert. Zudem können für kleine Datensätze auch händisch erzeugte Templates integriert werden.

Normalerweise werden Templates allerdings durch Einsetzen von Placeholdern aus einem Trainingsdatensatz im QALD-Format automatisch generiert. Placeholder sind dabei entweder Prädikate, Entitäten oder Klassen. Zur Beantwortung neuer Fragen wird eine strukturell äquivalente Frage gesucht, zu welcher bereits eine SPARQL-Query existiert. Die Idee hierbei ist, dass durch strukturell ähnliche SPARQL-Queries strukturell ähnliche natürlich-sprachliche Fragen beantwortet werden können.

Zur Transformation einer natürlich-sprachlichen Frage in eine SPARQL-Query müssen zunächst semantische Einheiten aus der Frage extrahiert und anschließend in die Query Templates eingefügt werden. Da für kleine Datensätze kein Klassifikator gelernt werden kann, werden in diesem Fall alle Templates mit den gefundenen semantischen Einheiten gefüllt. Für ausreichend große Datensätze wird ein Klassifikator zur Ermittlung des korrekten Templates verwendet. Hier wird semantische Ähnlichkeit als Graphisomorphismus betrachtet, da SPARQL eine Graph-basierte Anfragesprache ist.

Alle isomorphen Graphen werden dann einer Klasse mit einem Graph-Pattern für jedes Query-Template zugeordnet. Zur Zuordnung einer neuen Frage zu einer Klasse kann ein stochastisches Klassifikationsmodell verwendet werden. Dazu ist es notwendig, eine Frage in einen Feature-Vektor zu transformieren. Insgesamt werden sowohl semantische Features, welche inhaltliche Aspekte einer Frage repräsentieren, als auch syntaktische Features, die die Struktur einer Frage abbilden, verwendet.

Die aus den Trainingsdaten extrahierten Fragevektoren werden für das Lernen eines Klassifikationsmodells verwendet. Für das Trainieren und Evaluieren verschiedener Klassifikationsmodelle wurde das Java-Framework WEKA<sup>14</sup> eingesetzt, da es auch für Nicht-Domänen-Spezialisten leicht zu bedienen ist. Basierend auf der Evaluation kann dann das beste Machine-Learning-Modell ausgewählt werden. Für die letztendliche Beantwortung einer Frage muss

---

<sup>13</sup> <https://github.com/dice-group/TeBaQA>

<sup>14</sup> <https://www.cs.waikato.ac.nz/ml/weka/>

abschließend das durch den Klassifikationsalgorithmus identifizierte Template mit den gefundenen Klassen, Prädikaten und Entitäten gefüllt werden. Hierbei können auch bei der Klassifikation durch Permutation der gefundenen Entitäten mehrere mögliche Anfragen entstehen.

Deshalb wird abschließend ein regelbasiertes Ranking auf allen entstandenen Queries generiert, welches die Verlinkung der gefundenen Entitäten sowie deren Verbalisierung (RDF-Labels) verwendet. Falls keine Klassifikation eingesetzt wird, funktioniert das Ranking analog für alle generierten Templates. TeBaQA wurde zunächst auf dem großen QA-Datensatz QALD-9 getestet. Die Ergebnisse<sup>15</sup> zeigen, dass TeBaQA ein QALD-F-Measure von 0,35 erreicht, bei einer Precision von 0,27 und einem Recall von 0,27. Eine weitere Fehleranalyse ergibt, dass einfache Fragen wie »How much is the total population of the European Union?« in den meisten Fällen korrekt beantwortet werden können. Komplexere Fragen wie »What other books have been written by the author of The Fault in Our Stars?« sind jedoch schwerer zu beantworten, da ihre Struktur sehr viel umfassender ist.

Zudem wurde TeBaQA auf dem für das LIMBO-Projekt generierten QA-Benchmark mit 50 Fragen evaluiert, der mit dem Benchmark-Kuratierungssystem QUANT<sup>16</sup> erstellt wurde. Da bislang kein ausreichend großer Datensatz für das Training von QA-Systemen auf Mobilitätsdaten existiert, wurden manuell erstellte Templates ohne Klassifikation verwendet. TeBaQA erreicht hier ein QALD-F1-Maß<sup>17</sup> von 0,49 bei einer Precision von 0,32 und einem Recall von 0,33. Im Rahmen des Use Cases Suchmaschine auf Mobilitätsdaten wurde das Question-Answering-System TeBaQA zudem um eine Stichwortsuche erweitert.

## LIMBO-Pipeline

Neben dem Metadatenkatalog werden auf der öffentlichen GitLab-Instanz unter dem Gruppennamen LIMBO-Project<sup>18</sup> auch die LIMBO-Datensammlungen sowie dazugehörige Softwarewerkzeuge zur kontinuierlichen Pflege und Qualitätskontrolle der Daten gehostet. Diese Softwarewerkzeuge sind im Zuge der Entwicklung eines automatisierten Datenbereitstellungs-Prozesses (LIMBO-Pipeline) entstanden und folgen den Prinzipien der kontinuierlichen Integration, die bereits standardmäßig in der Softwareentwicklung angewendet werden. Die Übertragung dieser Prinzipien auf das Datenmanagement ist ein aktiver Forschungszweig im Bereich der Wissensgraph-Technologien, für den im LIMBO-Projekt neuartige Ansätze erarbeitet werden konnten. So ist es bei der Entwicklung der LIMBO-Pipeline gelungen, die folgenden Schritte der Bereitstellung von LIMBO-Daten stärker zu automatisieren und miteinander zu vernetzen:

- **Initialisierung der LIMBO-Laufzeitumgebung:** Durch die Speicherung der Daten in Repositorien des Git-Versionsverwaltungssystems können Änderungen jederzeit nachverfolgt werden. Daten-Eigentümer (engl. Owner) legen neue Repositorien mithilfe eines Software-Skriptes an. Auf diese Weise werden die entsprechenden Werkzeuge zur Publikation der LIMBO-Daten automatisch bereitgestellt. Dies sind:
  - dataset-includes: Ordner mit Build-Werkzeugen zur Durchführung von Operationen auf den Daten.
  - gitmodules: Datei, die dataset-includes als Submodul registriert.
  - .gitlab-ci.yml: Datei, die für Änderungen auf Datenebene automatische Tests triggert, die auf dem Gitlab-Server ausgeführt werden.<sup>19</sup> Die Funktionsweise dieser Art der

---

<sup>15</sup> <http://gerbil-qa.cs.upb.de:8080/gerbil/experiment?id=202010140006>

<sup>16</sup> <https://github.com/dice-group/QUANT>

<sup>17</sup> <http://gerbil-qa.cs.upb.de:8080/gerbil/experiment?id=202011230001>

<sup>18</sup> <https://gitlab.com/limbo-project>

<sup>19</sup> <https://gitlab.com/limbo-project/dataset-includes/-/blob/master/.gitlab-ci-template.yml>

Qualitätssicherung wird im Punkt Daten-Tests genauer beschrieben. Die Ergebnisse (0 Fehler = Erfolg, sonst Misserfolg) werden als Bild-Kennzeichen (engl. Badge) dem Repository auf der entsprechenden Gitlab-Instanz hinzugefügt. Die detaillierten Testprotokolle im RDF-Format können beim Klicken auf den Badge abgerufen werden.<sup>20</sup> Auf Grundlage der Test-Ergebnisse kann der Eigentümer entscheiden, ob die Datensammlung mit einer aktuellen Versionsnummer versehen und veröffentlicht werden soll.

- config.sh: Konfigurationsdatei, in der elementare Umgebungsvariablen (Version, Quell-Distribution des mCLOUD-Portals) für die Daten-Veröffentlichung gespeichert werden.
- Makefile: Datei, die auf das master-Makefile des dataset-includes Ordners referenziert.

Ausgehend von diesen initialen Konfigurationen zur Ausstattung eines Daten-Repositorys können Veröffentlichungsprozesse angestoßen werden. Dies kann entweder durch gekapselte Ausführung der für die Daten-Bereitstellung notwendigen Arbeitsschritte (teilautomatisch) oder mithilfe eines Softwaretools<sup>21</sup> erfolgen, das alle nicht-optionalen Arbeiten zur Veröffentlichung (Export Lifting-Portal, Datentransformation, Metadaten-Anreicherung, Publikation) automatisch nacheinander vollzieht (vollautomatisch). Das Vorhandensein beider Möglichkeiten der Daten-Publikation (teilautomatisch vs. vollautomatisch) ist wichtig, da es sein kann, dass ein Daten-Owner entweder einzelne Operationen auf den Daten ausführen möchte (z.B. Tests zur Identifizierung potentieller Fehler in den Daten) oder viele LIMBO-Daten effizient verarbeiten möchte (z.B. wenn mehrere Datensammlungen auf einmal veröffentlicht werden sollen). Für die Möglichkeit der vollautomatischen Veröffentlichung nutzt das Softwaretool ein Docker-Image, mit dessen Hilfe die LIMBO-Laufzeitumgebung bereitgestellt wird. Diese Umgebung enthält alle für die Ausführung der LIMBO-Pipeline benötigten Softwarewerkzeuge (z.B. SparqlIntegrate, cmemc, RDFUnit).

- **Export Lifting-Portal:** Hier werden die gewünschten LIMBO-Datensammlungen als RDF-Graphen aus dem Lifting-Portal exportiert. Der Export aus dem Lifting-Portal wurde durch die Integration des cmemc-Tools der eccenca GmbH in die LIMBO-Pipeline realisiert und kann durch Setzen von Umgebungsvariablen für den jeweils gewünschten LIMBO-Graphen vorgenommen werden.
- **Datentransformationen:** Hierbei handelt es sich um Daten-Operationen, die im Anschluss an die eigentliche Datentransformation, die bereits im eccenca Lifting-Portal stattgefunden hat, durchgeführt werden. Bei diesem sog. Post-Processing können kleinere Arbeiten zur Datenbereinigung vorgenommen werden (z.B. Ersetzung des Lifting-Portal-Namespaces mit dem LIMBO-Präfix).
- **Metadaten-Anreicherung:** Hier werden für die LIMBO-Datensammlung aussagekräftige Metadaten generiert und zu einer DCAT Beschreibung zusammengefügt.
- **Daten-Tests** (optional): Die Qualitätssicherung für LIMBO-Daten umfasst die folgenden Tests:
  - RDFUnit-Tests zur Überprüfung der Schemakonformität der Daten
  - Tests des CISS Quality Assurance Centers (CISS QAC), die auf der Inhaltsebene der Daten ansetzen und maßgeschneiderte Überprüfungen ausführen.
- **Linking** (optional): Um LIMBO-Daten untereinander mit verschiedenen Properties (z.B. owl:sameAs oder geonames:nearBy) zu verlinken, kann ein automatischer Prozess<sup>22</sup> ausgeführt werden, mit dessen Hilfe die Extraktion aktueller Datensammlungen und

<sup>20</sup> s. z.B. <https://gitlab.com/limbo-project/strassen-koeln>

<sup>21</sup> <https://gitlab.com/limbo-project/dataset-includes/-/blob/master/lif-n-release.sh>

<sup>22</sup> <https://gitlab.com/limbo-project/dataset-includes/-/blob/master/master/pipeline-linking.sh>

Linking-Konfigurationen aus dem Metadatenkatalog erfolgt. Die ermittelten Linking-Konfigurationen dienen dann der LIMES-Engine, die von der Universität Paderborn im LIMBO-Projekt weiterentwickelt wurde, als Grundlage für die automatische Erzeugung der Linksammlung. Sind einmal entsprechende Links generiert worden, können diese - analog zur Veröffentlichung der eigentlichen LIMBO-Daten - durch Nutzung der LIMBO-Pipeline auf Gitlab publiziert werden.

- **Daten-Publikation:** Zur Publikation der LIMBO-Daten auf der Gitlab-Instanz werden folgende Operationen durchgeführt:
  - *Durchführung des Release-Prozesses:*
    - Commiten der Releasedaten (Daten + DCAT-Eintrag)
    - Anlegen des Release-Tags, der dem aktuellen Stand der veröffentlichten Daten zugewiesen wird
  - *Zurückrollen des Release-Prozesses* (optional im Fehlerfall): Löschen des temporär angelegten Release-Branche sowie der Konfigurationsdatei für den Release-Prozess
  - *Deployment des DCAT-Eintrags:* Hinzufügen des erstellten DCAT-Eintrages zum Metadatenkatalog

## Metadaten-Management

Es wurde ein Verfahren zur automatischen Erstellung und Aktualisierung des LIMBO-Datenkataloges entwickelt. Die für diese Vorgänge relevanten Softwaretools können im GitLab-Repository des LIMBO-Metadatenkatalogs eingesehen werden.<sup>23</sup> Der Datenkatalog erfasst sowohl die Metadaten des mCLOUD-Portals als auch die Beschreibungen für LIMBO-Daten. Auf diese Weise können sich Interessierte über den jeweils aktuellen Stand vorhandener Sammlungen im Bereich der Mobilitätsdaten informieren. Die im DCAT-Format bereitgestellten Daten des mCLOUD-Portals (siehe z.B. die RDF-Beschreibung für einen Datensatz über Lichtsignalanlagen in Rostock aus der mCLOUD)<sup>24</sup> werden mithilfe eines Softwaretools in regelmäßigen Abständen abgefragt und durch Traversierung von HTTP-Links der Einzelseiten automatisiert gecrawlt. Diese Funktionalität wurde mit SparqlIntegrate sowie dem Make-Automatisierungstool realisiert.<sup>25</sup> Zur Serialisierung und gebündelten Speicherung dieser Daten wurden zudem Operationen für Daten-Transformationen konfiguriert, um die Informationen datentypkonform, mit eindeutigen Identifiern und dem LIMBO-Präfix zu versehen sowie von irrelevanten Metadaten (z.B. Crawling-Daten) zu bereinigen.

Neben den Beschreibungen der mCLOUD-Daten enthält der Datenkatalog auch Metadaten für Datensammlungen, die im LIMBO-Projekt entstanden sind. Diese Metadaten werden bei der Ausführung der Bereitstellungspipeline **(teil-)automatisch** im DCAT-Format erzeugt. Um diese Art der **Datenanreicherung** zu ermöglichen, wurde das Command Line Interface SparqlIntegrate so erweitert, dass Metadaten auf Basis nacheinander ausführbarer SPARQL-Abfragen erstellt werden können. Folgende Funktionalitäten wurden neu konzipiert und implementiert:

- Möglichkeit zum Übergeben und Aufrufen von Umgebungsvariablen zur parametrierbaren Ausführung von SPARQL-Abfragen
- Kommandozeilen-basierte Ausgabe der Ergebnisse von SPARQL-Abfragen im JSON-Format
- Kommandozeilen-basierte sequenzielle Abarbeitung von SPARQL-Abfragen auf RDF-Dateien sowie Serialisierung der Daten-Transformationen ins RDF-Format

---

<sup>23</sup> <https://gitlab.com/limbo-project/metadata-catalog>

<sup>24</sup> <https://www.mcloud.de/export/datasets/da57845f-ae51-41ca-944a-8f7beb132e8f>

<sup>25</sup> <https://gitlab.com/limbo-project/dataset-includes/-/blob/master/master/Makefile>

- Einlesen, Transformieren und Ausgeben von Named Graphs des RDF-Standards

Die konfigurierbare automatische Anreicherung von Metadaten wurde mithilfe von SparqlIntegrate und dem Makefile-Automatisierungstool implementiert. Die folgende Tabelle zeigt, dass die Metadaten-Anreicherung über verschiedene Methoden erfolgen kann.

Methode	Metadaten (Beispiele)
Festlegung in der Konfigurationsdatei (vom Daten-Eigentümer)	<ul style="list-style-type: none"> <li>- Name der LIMBO-Datensammlung</li> <li>- Versionsnummer (initial)</li> <li>- Quell-Datensammlung</li> </ul>
Automatische Generierung anhand der Daten-Eigenschaften	<ul style="list-style-type: none"> <li>- lokaler Identifier durch Auslesen des Dateinamens</li> <li>- Anzahl der enthaltenen Tripel</li> <li>- Entitätserkennung in der Beschreibung der Datensammlung</li> </ul>
Automatische Generierung anhand des Release-Prozesses	<ul style="list-style-type: none"> <li>- Download-Links</li> <li>- Versionsnummer</li> <li>- Release-Datum</li> </ul>

Im Zuge der automatischen Metadaten-Anreicherung können auf Basis der textlichen Beschreibung von Datensammlungen z.B. thematisch passende Schlagwörter vergeben werden. Für diesen Zweck wurde das Softwaretool KINDEX<sup>26</sup> erstellt, das durch Abfragen der Webservices DBpedia Spotlight, Wikidata<sup>27</sup> und LOBID GND<sup>28</sup> automatisch geeignete Deskriptoren der Gemeinsamen Normdatei (GND)<sup>29</sup> ermittelt (s. folgende Abbildung). Zudem ist das Tool so generisch konfiguriert, dass es für die Zuweisung weiterer Schlagwörter aus anderen Thesauri, wie z.B dem Standard-Thesaurus Wirtschaft<sup>30</sup> verwendet werden kann.

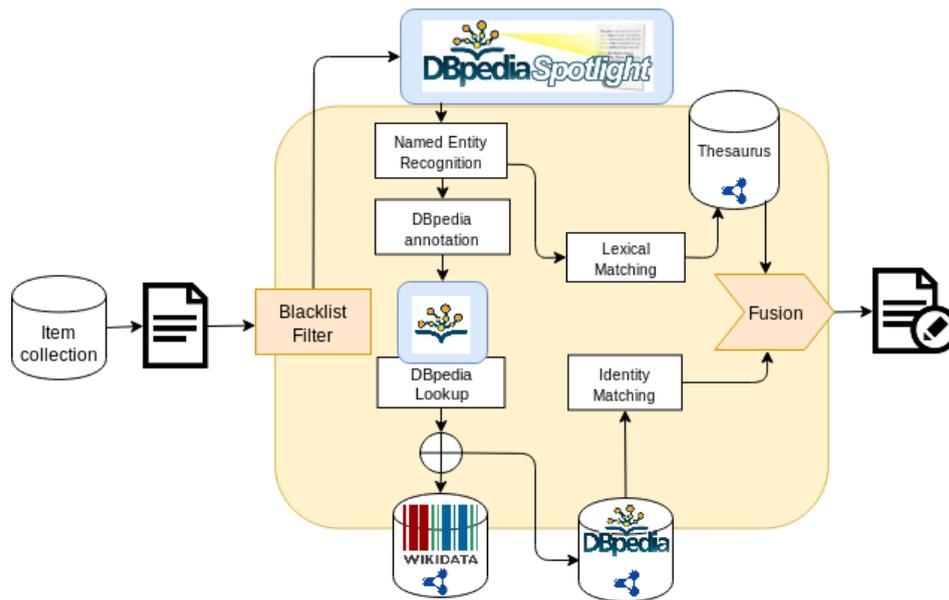
<sup>26</sup> <https://gitlab.com/limbo-project/keyword-indexing>

<sup>27</sup> <https://query.wikidata.org/>

<sup>28</sup> <https://lobid.org/gnd>

<sup>29</sup> [https://www.dnb.de/DE/Professionell/Standardisierung/GND/gnd\\_node.html](https://www.dnb.de/DE/Professionell/Standardisierung/GND/gnd_node.html)

<sup>30</sup> <https://zbw.eu/stw/version/latest/about.de.html>



Sowohl durch die automatische Metadaten-Anreicherung für LIMBO-Daten als auch im Rahmen der Ausführung von mCLOUD-Crawlings entstehen neue Beschreibungen für Datensammlungen. Sobald derartige DCAT-Daten zum Repository des Metadatenkataloges hinzugefügt werden, startet ein weiterer automatischer Prozess, der alle vorhandenen LIMBO- und mCLOUD-Metadaten zu einem Gesamtverzeichnis zusammenfügt, deren syntaktische Korrektheit überprüft und die LIMBO-Daten in einem LOD-Cloud-Diagramm visualisiert. Der Metadatenkatalog kann außerdem über einen SPARQL-Endpunkt<sup>31</sup> und einen Pubby-Browser<sup>32</sup> abgefragt werden.

## Orientierungs-App(s) für Sehgeschädigte

Aufgabe der appPlant GmbH als Anwendungspartner war die Validierung der Projektergebnisse durch Entwicklung einer prototypischen Anwendung auf Basis der gelifteten und gelinkten Daten. Geplant war zunächst eine Applikation für den Anwendungsfall Prädiktive Logistik 4.0. Im Laufe des Projekts stellte sich jedoch ein anderer, bis dahin in einem unternehmensinternen Projekt verfolgter Anwendungsfall als sinnvoller heraus: (Innenraum-)Orientierung für Sehgeschädigte.

Dabei wurde an folgenden technischen Stand angeknüpft:

- Standortbestimmung mittels externer Sensorik:
  - BLE (Bluetooth Low Energy)
  - WIFI (Wireless Fidelity)
  - WLAN (Wireless Local Area Network)
- Standortbestimmung mittels:
  - SLAM (Simultaneous Localization and Mapping)
  - IMU (Inertial Navigation Systems) mobiler Endgeräte (z.B. Geschwindigkeitssensoren)
  - AR (Augmented Reality), im Fall der mobilen Endgeräte v.a. Nutzung von Bilddaten, die durch die Kamera generiert werden
- Sprachverarbeitungssoftware

Im Projekt wurden zwei Prototypen als mobile Assistenzsysteme für Sehgeschädigte entwickelt. Diese stellen zwei Seiten einer Medaille dar, da Sehgeschädigte auch Unterstützung beim Übergang zwischen

<sup>31</sup> <https://metadata.limbo-project.org/sparql>

<sup>32</sup> <https://metadata.limbo-project.org/Catalog>, <https://data.limbo-project.org/Overview>

“draußen” und “drinnen” benötigen:

1. ALeite (Indoor-Orientierung)
2. BlindSpot (Outdoor-Orientierung)

Insbesondere BlindSpot ist im Hinblick auf die Nutzung gelifteter und gelinkter Datensätze und damit auf die Erfüllung der Aufgabenstellung innerhalb des Projekts näher zu beschreiben:

Implementiert sind in der Beta-Version für iOS:

a) User Interface speziell für Sehgeschädigte

- mit Sprachsteuerung
- mit vereinfachter Gesten-Steuerung

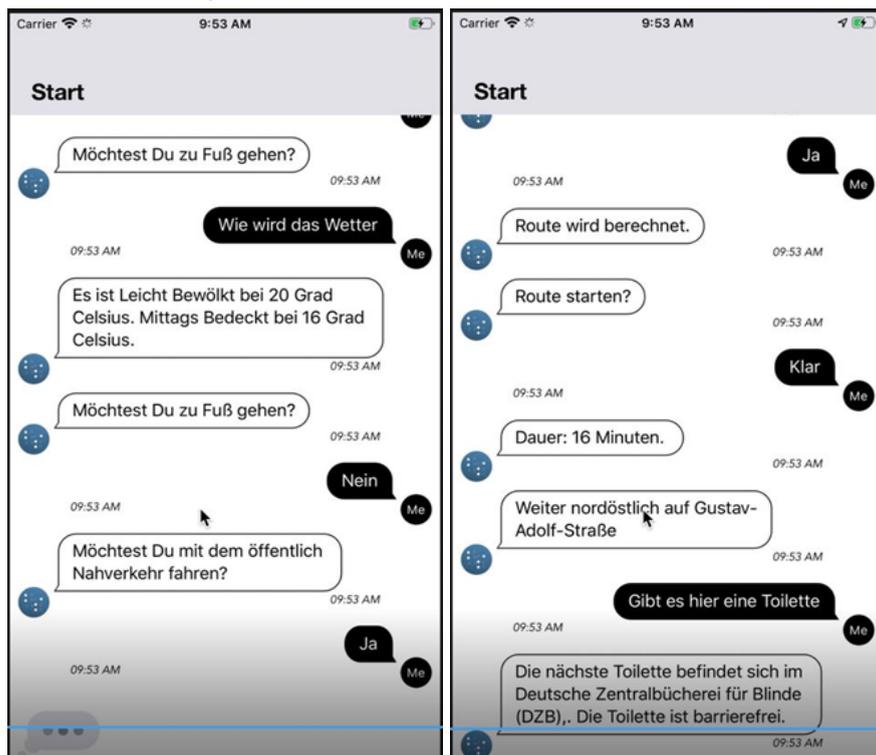
b) Outdoor-Orientierung und Navigation für Sehgeschädigte

- Finden eines Routen-Ziels (OpenStreetMap-Daten)
- Auswahl eines Verkehrsmittels (ÖPNV- und kommunale Daten)
- Abfrage und Ausgabe von Informationen zu Wetterverhältnissen (DWD-Daten)
- Ausgabe von Informationen zu interessanten Punkten in der Umgebung des Standortes (OpenStreetMap-Daten)
- Routen-Berechnung
- Routen-Beschreibung (Dauer, Fußwege, ÖPNV-Verbindung, spezifische Zusatzinformationen für Sehgeschädigte wie z.B. Informationen zu taktilen Wegführungen, barrierefreien Ampeln)

c) Rudimentäre Indoor-Orientierung

- Ausgabe von Informationen zu Gebäuden (Daten des Behindertenverbands Leipzig e.V.)

Die Interaktion des Nutzers erfolgt primär per Sprachein- und -ausgabe. Dennoch wird der Interaktionsverlauf mittels einer Speech-to-Text-Funktion zusätzlich visualisiert, was für Sehgeschädigte z.B. in Situationen, in denen sie nach Hilfe fragen möchten, hilfreich sein kann. Eine typische Interaktion mit der App sieht verschriftlicht folgendermaßen aus:



Der im Rahmen des LIMBO-Projekts fertiggestellte Prototyp zur Innenraum-Orientierung für Sehgeschädigte ALeite, ist die Ausgangsbasis für ein Folgeprojekt, in dem die Integration der Orientierungs-App mit der Datenbasis BIM (Building Information Modeling) angestrebt wird. Während der Durchführung des LIMBO-Projekts wurden, auch durch Hinweise des Zuwendungsgebers, entscheidende Fortschritte auf dem mit ALeite beschrifteten Gebiet bekannt: Das Fraunhofer-Institut FOKUS hat die App "everguide Indoor Navigation" entwickelt, die derzeit in einer Beta-Version für Android vorliegt. Der hierbei verfolgte technologische Ansatz wurde allerdings zugunsten der Einbindung von BIM von der appPlant GmbH nach Abschluss von ALeite verworfen.

Durch Integration des zweiten, zum Ende der Projektlaufzeit fertiggestellte Prototyps für Outdoor-Orientierung, BlindSpot, mit iBN, dem Nachfolger von ALeite, wird das Ziel verfolgt, Sehgeschädigten einen nahtlosen Übergang aus der städtischen Umgebung in ein Gebäude zu ermöglichen.

BlindSpot an sich wird zudem weiterentwickelt, u.a. ist eine Zusammenarbeit mit der Nahverkehrsservice Sachsen-Anhalt GmbH und Sozialhelden e.V. (accessibility.cloud) geplant. Langfristiges Ziel nach einer Testphase mit Mitgliedern des Blinden- und Sehbehindertenverbandes Sachsen e.V. Leipzig ist ein Ausrollen in der Stadt Leipzig und ggf. benachbarten Städten.

## VoCoL

Eine der Aufgaben des Fraunhofer IAIS war die Bereitstellung einer kollaborativen Ontologie-Bearbeitungs Umgebung. Diese wurde in Form von VoCol - An Integrated Environment for Collaborative Vocabulary Development<sup>33</sup> dem Projekt Konsortium bereitgestellt und kontinuierlich während des Verlaufes des Projektes weiterentwickelt. Neue Anforderungen, die im Laufe des Projektes aufgekomen sind, wurden dabei eingepflegt. Die initiale Implementierung nutzt dabei Quelldateien, die auf Github<sup>34</sup> gehostet werden. Zentral ist dabei die Versionskontrolle, welche auf die im Projekt verwendeten Vokabulare angewandt wurde und eine nahtlose Verfolgung der einzelnen Änderungen ermöglicht. Nebenher ist VoCol in der Lage, syntaktische und semantische Fehler zu erkennen. Sobald diese erkannt werden, kann der Nutzer diese beheben. Die im LIMBO-Projekt getätigten Weiterentwicklungen sind dabei einerseits in das inzwischen abgelöste VoCol eingeflossen, finden sich aber auch im Nachfolger VOCOREG - VoCol Service<sup>35</sup> wieder, welcher im nachfolgenden Screenshot dargestellt wird.

---

<sup>33</sup> <https://vocol.iais.fraunhofer.de/ids/>

<sup>34</sup> <https://github.com/vocol/mobivoc/>

<sup>35</sup> <https://vocoreg.com>

MOBI VOC

[Home](#) [Editing](#) [Documentation](#) [Visualization](#) [Querying](#) [Evolution](#) [Analytics](#) [Validation](#) [Branch](#) [develop](#)



## Open Mobility Vocabulary

MobiVoc is an open vocabulary for future-oriented mobility solutions.

New mobility concepts and better data networking are both crucial factors for global economic development. To invent innovative and sustainable mobility concepts, new data-based value-added services are required.

Our goal is to significantly improve the data mobility between all stakeholders by providing a standardized vocabulary using Semantic Web technologies and ontologies. For the open vocabulary covering various mobility aspects we use RDF (Resource Description Framework) - a recommended specification of the World Wide Web Consortium (W3C) and the so-called lingua franca for the integration of data and web. We invite everyone who is interested to join our MobiVoc initiative and to participate in the development of the Open Mobility Vocabulary.

MobiVoc was extended in the following research projects: LIMBO, bloTope.

### Ontology Metadata

rdftype	<a href="http://www.w3.org/2002/07/owl#Ontology">http://www.w3.org/2002/07/owl#Ontology</a>
	<a href="http://purl.org/voccommons/voaf#Vocabulary">http://purl.org/voccommons/voaf#Vocabulary</a>
rdfscomment	MobiVoc is a standardized vocabulary (also known as 'schema', 'Data Dictionary' or 'ontology') for mobility data. Currently, the following topics are covered: Electric Charging Points, Parking Facilities, Highway Roadworks sites (Datex II)@en
rdfs:isDefinedBy	<a href="http://schema.mobivoc.org/#">http://schema.mobivoc.org/#</a>
rdfs:label	MobiVoc: Open Mobility Vocabulary@en
contributor	<a href="http://sebastian.tramp.name">http://sebastian.tramp.name</a> <a href="http://purl.org/net/clange">http://purl.org/net/clange</a>

Data Protection Imprint © 2021

Die inzwischen etablierte Nachfolge-Plattform wird als zentrale Infrastruktur für Data Space Projekte unter der International Data Spaces Association weiterverwendet. Unter anderem werden die im Mobilitätsbereich relevanten Anpassungen an die integrierte Ontologie Kontrolle im Zuge des mFund Förderprojektes Mobility Data Spaces eingebracht, um eine automatische Datenintegration der Daten im Daten Marktplatz zu ermöglichen (September 2021). Ziel ist dabei die Wiederverwertung der bereits entwickelten Ontologien auch für Folgeprojekte.

## Anwendungsszenarien

### Suche auf Mobilitätsdaten

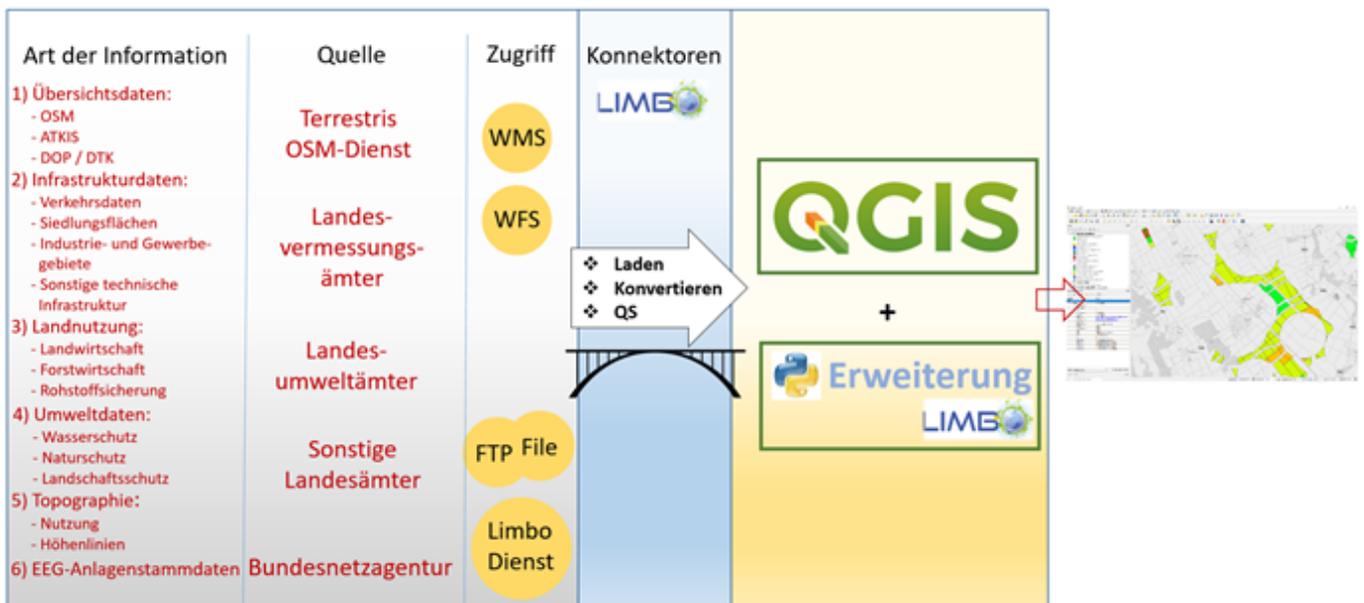
Zur Umsetzung der Suche wurde das in AP 6 entwickelte Fragebeantwortungssystem TeBaQA verwendet. Zudem wurden die im LIMBO-Projekt entstandenen Datensätze indiziert. Gleichzeitig wurde die entwickelte Stichwortsuche in das TeBaQA-Interface<sup>36</sup> integriert, um gleichzeitig eine Stichwortsuche und die Fragenbeantwortung zu ermöglichen.

### Weißflächenkartierung

Ein praktisches Anwendungsbeispiel im LIMBO-Projekt ist die weitestgehend automatisierte Auswertung von offenen Geodaten zur Analyse und Planung von neuen Windkraftanlagen (WKA).

Der gesamte Prozess läuft – vereinfacht dargestellt – in 3 Phasen ab:

1. Identifizieren der benötigten Datenbestände, anhand länderspezifischer Vorgaben.
2. Weißflächenkartierung zur Identifizierung von grundsätzlich in Frage kommenden Flächen, vorgegeben beispielsweise durch gesetzliche Abstände zu Gebäuden etc. (s. Abb. 1).
3. Potentialanalyse mit Schwerpunkt auf
  - Windgeschwindigkeiten
  - Transportkosten
  - Zuständige Genehmigungsbehörden



Schema Weißflächenkartierung

Für die Orientierung können Übersichtskarten (OSM, ATKIS oder DOP), die über WebMapServices bereitgestellt werden, Verwendung finden.

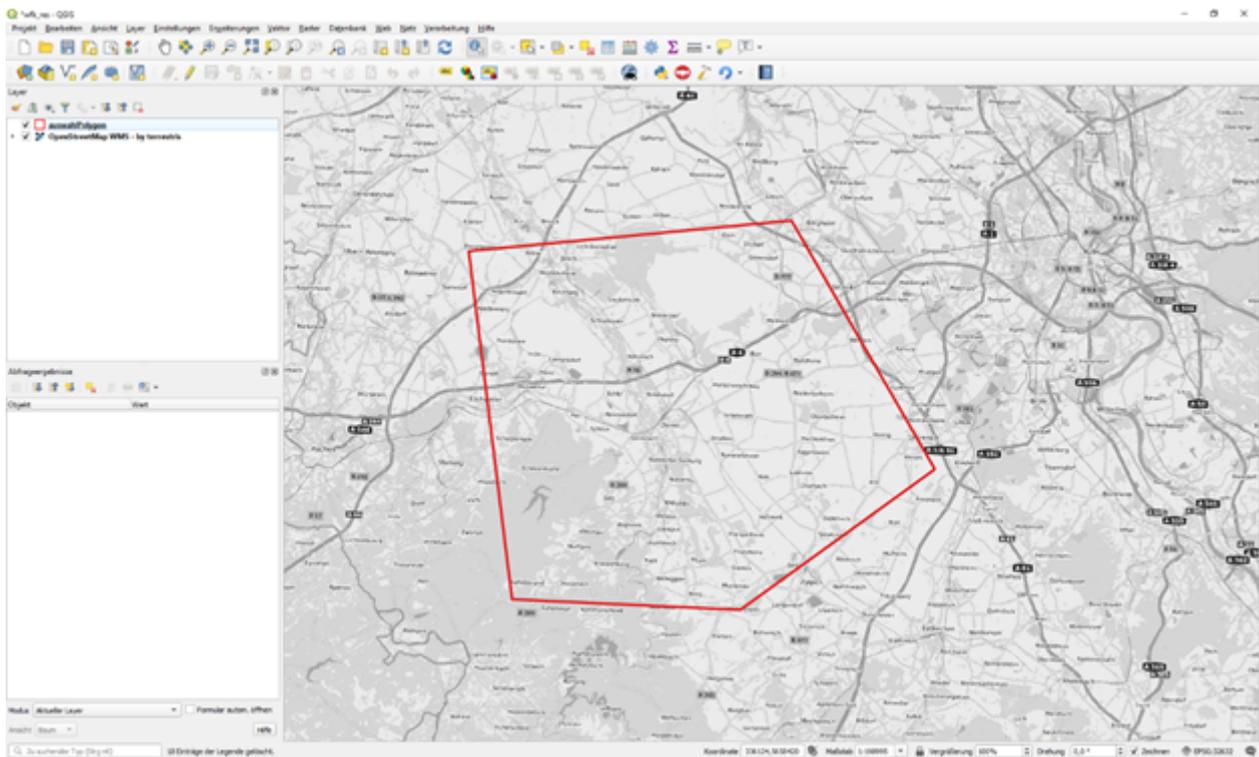
Bei der Weißflächenkartierung sind für jedes Bundesland unterschiedliche Regelungen zu beachten, hierunter fallen Ausschlusskriterien für 1) Verkehrsflächen, 2) Siedlungsflächen, 3) Industrie- und Gewerbe, 4) sonst. Technische Infrastruktur (auch Landesverteidigung) 5) Land- und Forstwirtschaft, 6) Rohstoffsicherheit, 7) Gewässerschutz 8) Arten und Biotopschutz, sowie 9) Landschaftsschutz (siehe obige Abbildung). Jede dieser Regelung definiert den Mindestabstand der zu planenden Windkraftanlage in Bezug auf die entsprechenden Kriterien. Die Regelungen wurden im Rahmen des

<sup>36</sup> <https://suche-limbo.demos.dice-research.org/>

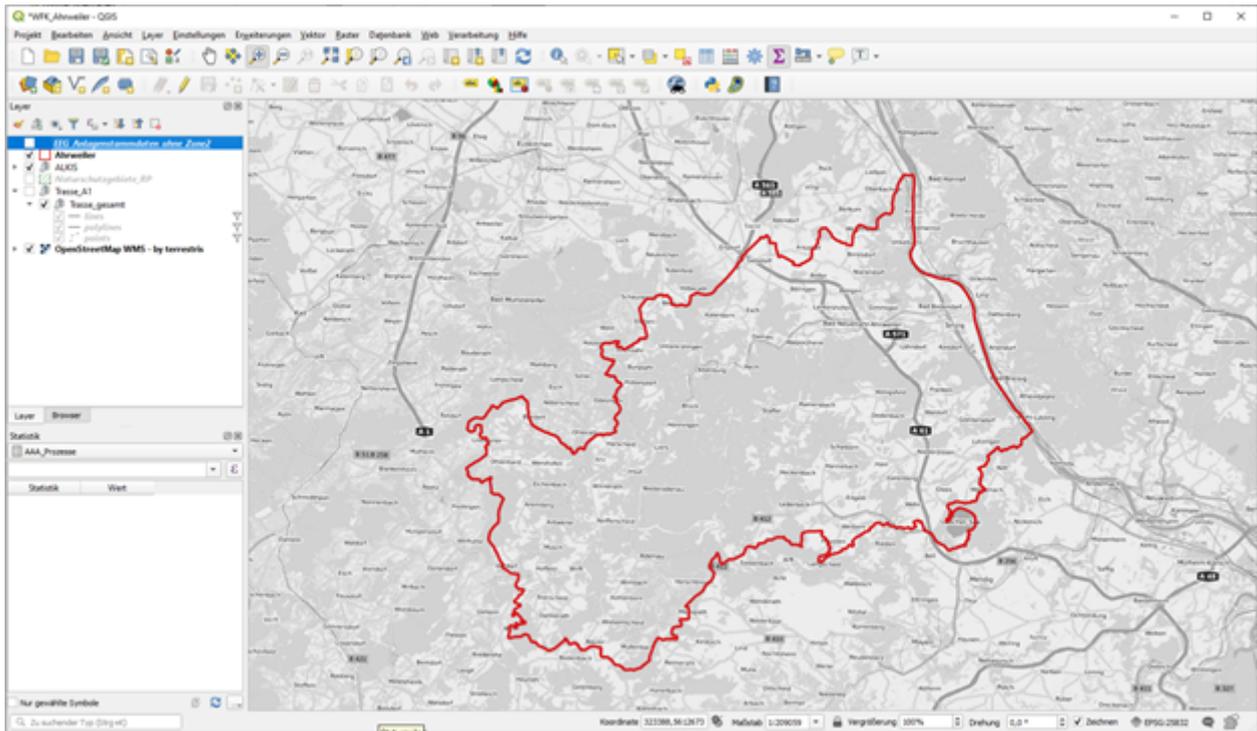
Projektes LIMBO gesammelt, um deren Beachtung bei der Integration und Planung gewährleisten zu können.

Seit 2017 sind die Geobasisdaten in NRW öffentlich verfügbar. Außerdem stellen weitere Landes- und Bundesbehörden, wie das LANUV und die Bundesnetzagentur relevante GIS Datenbestände zum Download bereit. Umwelt- und Forstdaten sind auch in RP kostenfrei zu bekommen. Die ALKIS-Daten müssen allerdings käuflich erworben werden. Leider sind Regionalpläne oder Bebauungspläne nicht zentral zugänglich, sondern ‚verstecken‘ sich auf den Webseiten der Kommunen und sind oft nur als PDF zugänglich, also nicht von Geoinformationssystemen direkt verarbeitbar.

Für die prototypische Durchführung des Planungsprozess‘ sind die beiden Gebiete Köln/Aachen (NRW, s. untenstehende Abbildung), sowie der Landkreis Ahrweiler (RP) betrachtet worden (s. untenstehende Abbildung). Ersteres liegt im Bereich der Braunkohlegrube Hambach, bei zweitem handelt es sich um das Gebiet der Heimatgemeinde von CISS TDI GmbH. Grundlage hierfür ist das Interesse des Landkreises an einer Weißflächenkartierung und dementsprechend an einem Demonstrator für Vorhaben in dieser Richtung.



Testgebiet 1 zwischen Köln und Aachen im Bereich des Tagebau Hambach



Testgebiet 2 Landkreis Ahrweiler

### Identifikation relevanter Daten und Quellen

Zum Zeitpunkt der Implementierung wurden folgende relevante Bestände unter Open Data identifiziert:

**Nordrhein-Westfalen:** ALKIS-Daten, Basis-DLM, DOM1, FFH-Gebiete, Nationalparke, Naturparke, Landschaftsschutzgebiete, Feuchtgebiete, Naturschutzgebiete, Vogelschutzgebiete, Windgeschwindigkeiten aus den COSMO-REA2 Daten des DWD, Open Street Map und EEG-Anlagen.

Als Quellen der aufgeführten Daten sind im Wesentlichen zu nennen:

- OpenGeodata.NRW ([www.opengeodata.nrw.de](http://www.opengeodata.nrw.de))
- OpenStreetMap arbitrary excerpt export (<http://osmaxx.hsr.ch>)
- Geofabrik ([www.geofabrik.de](http://www.geofabrik.de))
- Universität Bonn (<ftp://ftp.meteo.uni-bonn.de/pub/reana/COSMO-REA2>)
- Bundesnetzagentur ([www.bundesnetzagentur.de](http://www.bundesnetzagentur.de))

**Rheinland-Pfalz:** ALKIS-Daten, Basis-DLM, DOM1, FFH-Gebiete, Naturschutzgebiete, landesweit bedeutsame Kulturlandschaften (LahiKula), geschützte Laubholzbestände (Landesforsten Rheinland-Pfalz), zivile und militärische Richtfunkstrecken, Flächen für den Ausbau der A1 (Landesbetrieb Mobilität und Verkehr), FFH-Gebiete, Vogelschutzgebiete, Windgeschwindigkeiten aus den COSMO-REA2 Daten des DWD, Open Street Map ,EEG-Anlagen, ...

Als Quellen der aufgeführten Daten sind im Wesentlichen zu nennen:

- Landesamt für Vermessung und Geobasisinformation (<http://lvermgeo.rlp.de/de/geodaten/opendata/>)
- Landesforsten Rheinland-Pfalz (<http://www.wald-rlp.de>)
- Landesbetrieb Mobilität Rheinland-Pfalz (<http://www.lbm.rlp.de>)
- OpenStreetMap arbitrary excerpt export (<http://osmaxx.hsr.ch>)
- Geofabrik (<http://www.geofabrik.de>)
- Universität Bonn(<ftp://ftp.meteo.uni-bonn.de/pub/reana/COSMO-REA2>)
- Bundesnetzagentur (<http://www.bundesnetzagentur.de>)

Im Sinne der oben erwähnten Planungskomplexe sind diese überwiegend amtlichen Datenbestände

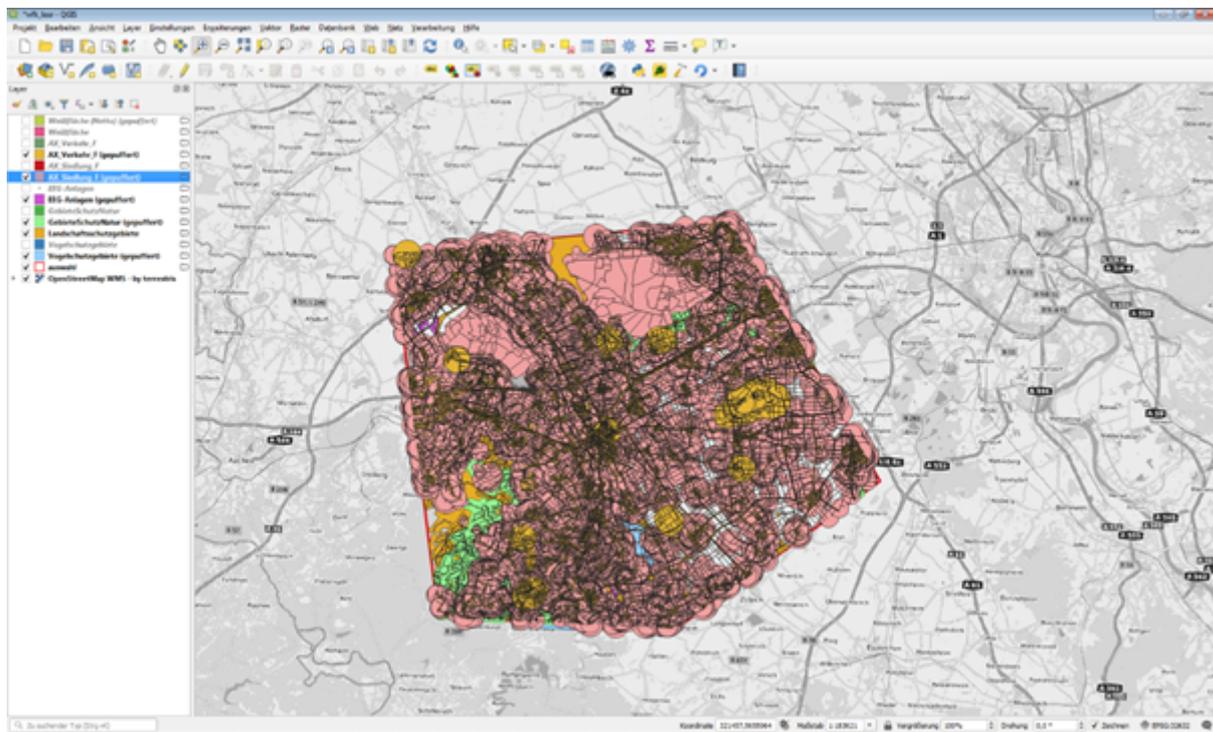
ausreichend, um Weißflächen für eine weitere Analyse zu identifizieren.

### Automatisierte Weißflächenkartierung

Für die Planung von Windkraftanlagen wurde ein QGIS Python-Plugin entwickelt, das eine automatisierte Weißflächenkartierung (WFK) durchführt.

Ziel dieser Entwicklung war eine intuitive-Lösung für eine WFK bereitzustellen. Das Tool kann dazu Vektorgrafiken aus unterschiedlichen Quellen mit unterschiedlicher Herkunft heranziehen. Das können beispielsweise ALKIS-Siedlungsflächen aus einem WFS, Naturschutzgebiete aus einer SHAPE-Datei oder WKT-Geometrien aus einem Excel-Format sein. Das Tool muss vor der Ausführung konfiguriert werden. Hier müssen die Herkunft, Format, Koordinatensystem, etc. der Daten angegeben werden und welche geometrische Funktion auf die Auswahlfläche angewendet werden soll. Weiterhin müssen die im jeweiligen Bundesland geltenden Abstandsregelungen (Puffer-Abstände) für Windkraftanlagen definiert werden.

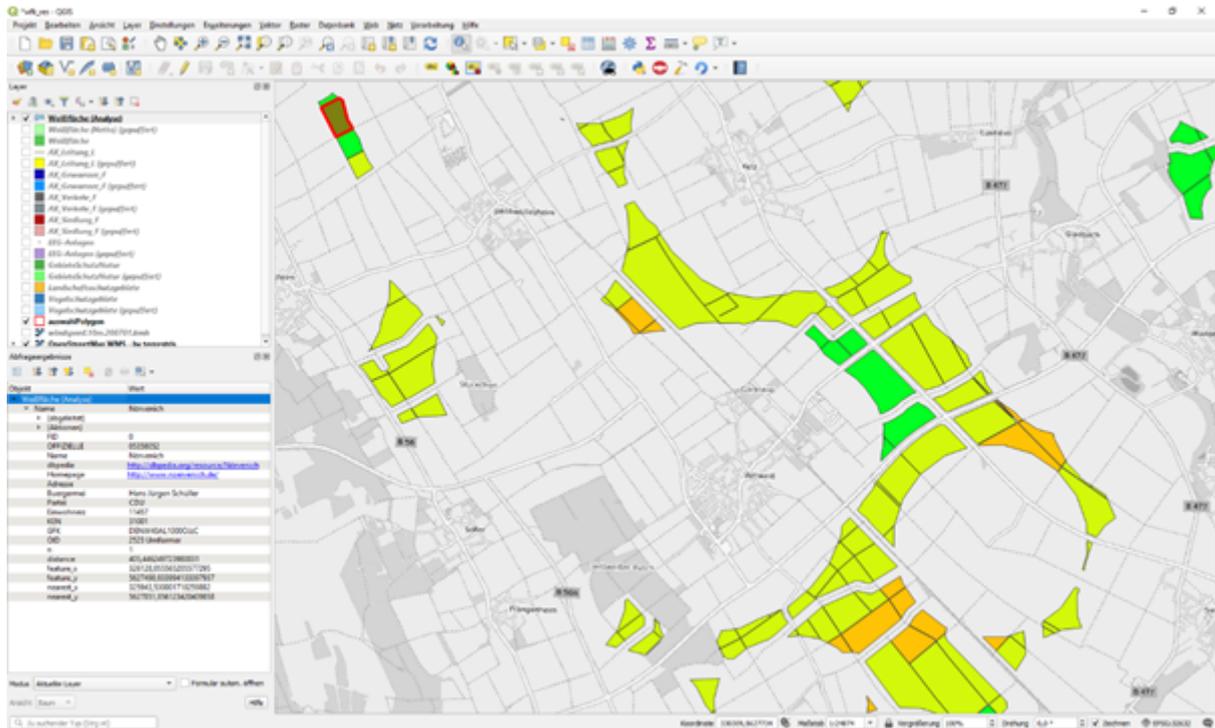
Nach Subtraktion aller definierten Flächen samt Pufferzonen von der Auswahlfläche verbleibt die sogenannte Weißfläche, also die Fläche, auf der nach gesetzlichen Vorgaben eine Windkraftanlage errichtet werden kann (nachfolgende Abbildung).



In transparent die 'übriggebliebenen' Bereiche, also die Weißflächen

### Potentialflächenanalyse

Diese Weißflächen können im Anschluss weiter automatisiert analysiert werden. Beispielsweise lassen sich Merkmale definieren, um die Weißflächen thematisch einzufärben. Dadurch treten planerische und infrastrukturell wichtige Kriterien, wie Beispielsweise die Distanz zum nächsten Umformer, direkt visuell hervor (nachfolgende Abbildung).



Weißflächen mit thematischer Einfärbung zum Kriterium 'Distanz zum nächsten Umformer' und Informationen zur Kommune

Zudem können LIMBO-Dienste zur weiteren Bewertung durch Zusatzinformationen verwendet werden. Implementiert sind Zusatzinformationen aus einem SPARQL-Endpoint, der Informationen zu den kommunalen Verwaltungsorganen aus dbpedia liefert (oben stehende Abbildung).

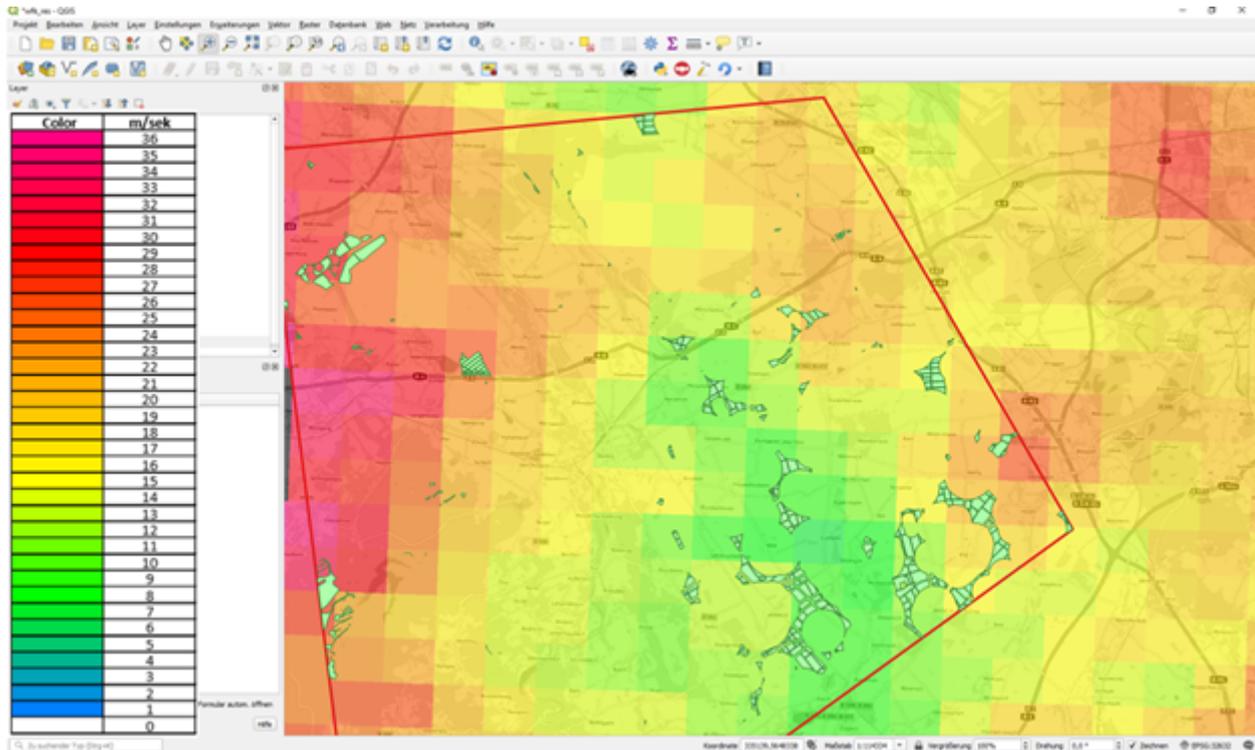
Das Tool bietet desweiteren die Möglichkeit eine Kostenberechnung für die Anlieferung der Anlagen vorzunehmen. Diese Kostenberechnung soll unterschiedliche Straßentypen berücksichtigen, aus denen jeweils ein charakteristischer Kilometerpreis resultiert. Auch kann festgestellt werden, ob der letzte Kilometer Zuweg zum Bauplatz schon vorhanden ist, oder noch gebaut werden muss, was natürlich die Kosten stark beeinflusst. Das Tool zur Routenplanung mit Kostenfaktoren wurde in Zusammenarbeit mit USU realisiert.

Für die Potentialflächenanalyse gilt besondere Aufmerksamkeit zudem den Reanalysedaten. Im Projekt LIMBO wurde gemeinsam mit den Projektpartnern (insbesondere dem Institut für angewandte Informatik, InfAI) ein Datenmodell, sowie eine Ontologie für die des DWD bereitgestellten COSMO-REA2/REA6 entwickelt und in den Planungsprozess von Windenergieanlagen oder –parks integriert (Abb. 7). Derzeit werden für die Berechnungsprogramme Reanalysedaten wie MERRA2 (kostenfrei) und EMD/Conwx (ERA5) verwendet. Die vom DWD generierten COSMO-REA2 Daten sind erst seit 2017 verfügbar und decken die Periode von 2007 – 2014 ab (Stand 31.12.2019). Seit dem Fertigstellen des derzeitigen Datenbestandes gab es keine Aktualisierung mehr. Dafür hat COSMO-REA2 mit seinem 2 km x 2 km dichten Netz an Probepunkten die mit Abstand genaueste Analyse. Eine Gegenüberstellung wichtiger Parameter findet sich in der nachfolgenden Tabelle.

	ConWx Data	NASA Merra-2	DWD COSMO-REA
<b>Räuml. Auflösung</b>	3 x 3 km	50 km	2 x 2 km
<b>Zeitl. Auflösung</b>	1 h	3 h	15 m
<b>Zeitl. Abdeckung</b>	> 20 years	> 30 years	2007 - 2014
<b>Aktualisierung</b>	Monatlich	Monatlich	-
<b>Kosten</b>	1500,00 €/Jahr	Frei	-
<b>Schnittstelle zu WindPro</b>	+	+	-

Vergleich Reanalyse-Daten am Markt

Die Tabelle zeigt die teils deutlich bessere räumliche und zeitliche Auflösung der Reanalysedaten, jedoch eine deutlich schlechtere zeitliche Abdeckung. Bei den Interviews mit potentiellen Windparkbetreibern und Projektieren sind diese Daten durchaus auf Interesse gestoßen, erlauben aufgrund ihrer kurzen zeitlichen Abdeckung jedoch nur einen eingeschränkten Mehrwert und sind in Konsequenz derzeit nicht als einzige Quelle verwertbar. Die Prozesse und der Großteil der Technologien, die in diesem Projekt entwickelt wurden, lassen sich jedoch im Nachgang des Projektes auf andere Datenquellen adaptieren, sodass dieses Ergebnis nicht als kritisch eingestuft wird.

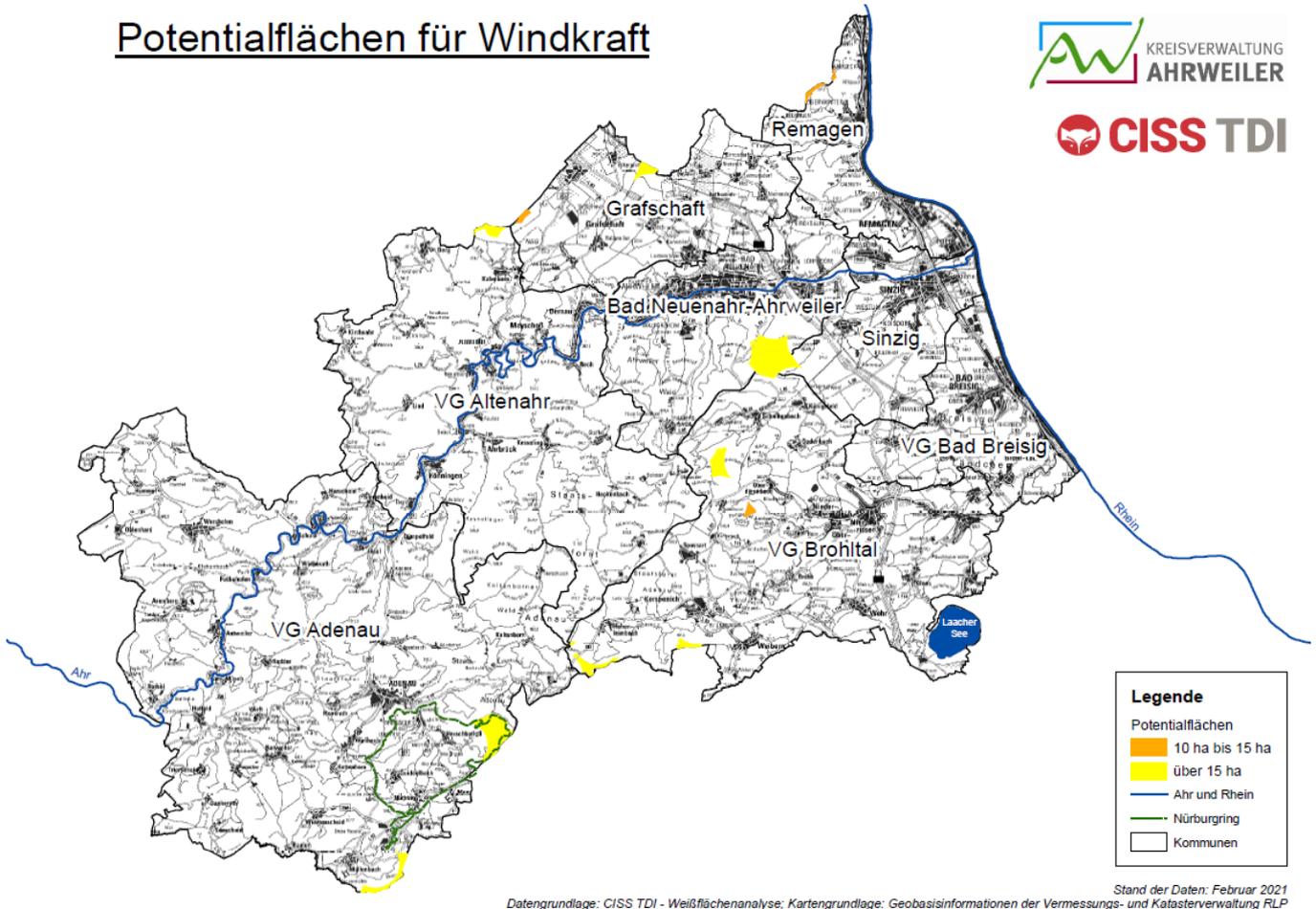


Auswahlgebiet mit Weißflächen und Windspeed aus COSMO-REA2  
Ausblick

Nachdem die Vorstellung eines frühen Prototyps des Weißflächenkartierungstools bei den Windenergietagen 2019 in Potsdam in der Branche auf reges Interesse stieß, ist auch eine Weiterentwicklung in Richtung Photovoltaik denkbar, bei der ganz eigene Abstandsregeln berücksichtigt werden müssen.

Für die Erstellung von Gutachten zur Anzahl von potentiellen Weißflächen in den Kommunen, werden neben dem Landkreis Ahrweiler weitere Probanden gesucht, mit denen das Tool nach Projektende praxisnah weiterentwickelt werden kann. Gemeinsam mit dem Landkreis Ahrweiler wurden Daten aus den unterschiedlichsten Bereichen (unter anderem ALKIS, Naturschutz, Landesplanung, Forst, etc.) ausgewertet. Ziel war es herauszufinden, wo es sich im Kreis Ahrweiler lohnt, Windkraftanlagen zu planen. Das Ergebnis dieser Prüfung zeigt das nachfolgende Bild.

## Potentialflächen für Windkraft



Nach Abschluss des Projektes wurde das QGIS-Plugin zur Weißflächenkartierung zur freien Verwendung veröffentlicht<sup>37</sup>.

### Intelligente Service-Unterstützung (Industrie 4.0)

Im Bereich der Anlagengüter und Produktionsmaschinen sind begleitende Services und Dienstleistungen ein zunehmender Umsatzträger und Abgrenzungsmerkmal im internationalen Wettbewerb. Ein erheblicher Anteil der Dienstleistungen wird dabei vor Ort erbracht. So müssen Servicetechniker nach wie vor Wartung und Service vor Ort betreiben, d.h. bestimmte Bauteile austauschen oder die Maschine reinigen. Eine Herausforderung stellt dabei die Einsatzplanung unter Berücksichtigung verfügbarer Informationen dar. Einen Schwerpunkt bilden hier vor allem die Verkehrs- und Klimadaten aus der mCLOUD. Eine Herausforderung stellt hier beispielsweise die Logistik von Servicekräften, Spezialwerkzeugen und Ersatzteilen unter Berücksichtigung von klimatischen oder verkehrstechnischen Änderungen dar.

Zunächst wurden relevante Datensätze identifiziert, die im Anwendungsfall verwendet werden sollen. Dabei ergaben sich zwei Kategorien von Daten: (Echtzeit-) Mobilitätsdaten und Auftragsdaten. Auftragsdaten beinhalten Informationen über geplante Servicetechniker-Einsätze (Kunde, Anschrift, Auftragsart, ...) aber auch Informationen über Servicetechniker (Fähigkeiten, vorherige Aufträge, ...). (Echtzeit-) Mobilitätsdaten beinhalten alle Informationen die einen Servicetechniker unterwegs im Auto beeinflussen (Stau, Baustellen, ...).

Aus dem Forschungsprojekt STEP wurde ein Datensatz zu Service Techniker Einsätzen in das LIMBO

<sup>37</sup> <https://plugins.qgis.org/plugins/weissflaechenkartierung/>

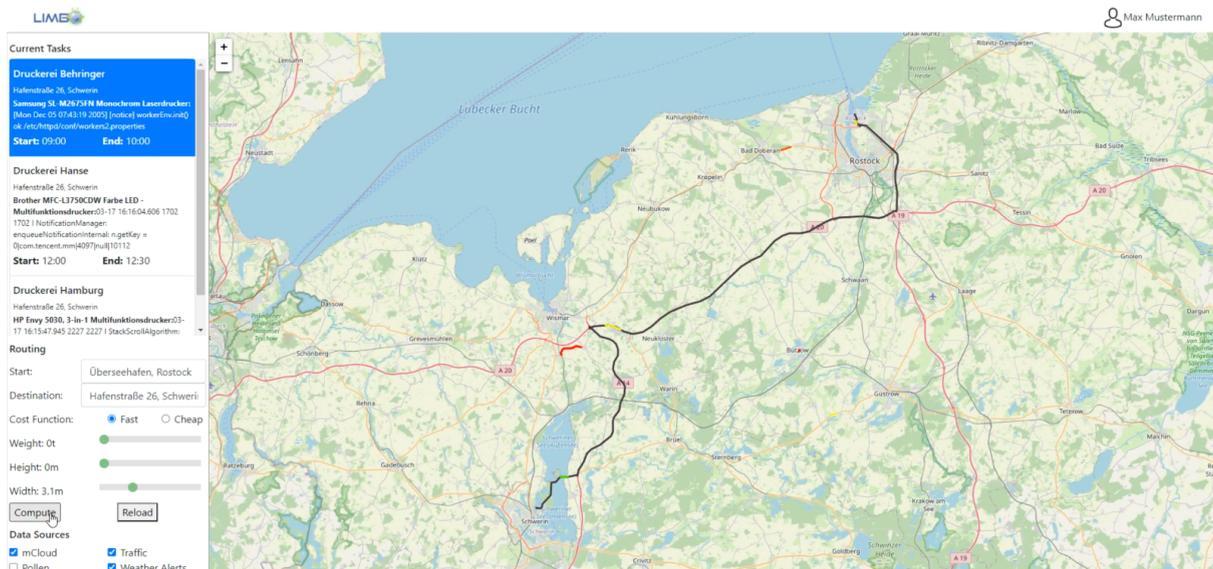
Projekt eingebracht der historische Service Aufträge eines deutschen Maschinenbauers enthält. Aus lizenzrechtlichen Gründen darf dieser Datensatz jedoch nicht im LIMBO Projekt verwendet werden. Da die Daten für die Umsetzung des Use-Cases Intelligente Service Unterstützung (Industrie 4.0) jedoch zwingend erforderlich sind, wurde ein Programm zur Simulierung von Service Techniker Einsätzen entwickelt. Dieses Programm erlaubt die Generierung von realitätsnahen Einsatzbeschreibungen mit dem Vorteil das keinerlei Lizenzen verletzt werden.

Mobilitätsdaten werden größtenteils aus der mCLOUD übernommen. Es wurden insgesamt 13 öffentlich zugängliche Datensätze aus der mCLOUD eingebunden. Da in der mCLOUD nur historische Daten verfügbar sind wurden zusätzlich drei Datensätze mit Echtzeitinformationen außerhalb der mCLOUD eingebunden. Abschließend wurde noch das Straßennetz von OpenStreetMap importiert. Benötigte mCLOUD Datensätze wurden über die LIMBO Plattform gelistet, bei Bedarf mit weiteren Geo-Informationen angereichert und als Linked-Data publiziert. Dadurch ist es nun möglich diese Daten für eine dynamische Routenberechnung zu verwenden und gleichzeitig graphisch auf einer Karte für einen Servicetechniker aufzubereiten.

Neben einer reinen Visualisierung werden die erhobenen Daten auch verwendet, um eine Routenplanung für die Anfahrt von Service-Technikern durchzuführen. Für die Routenberechnung werden zwei Arten von Anpassungen vorgenommen:

- Die durchschnittliche Geschwindigkeit einer Straße wird reduziert, um ein erhöhtes Verkehrsaufkommen oder schlechtes Wetter zu simulieren.
- Eine Straße wird aus dem Routing Graphen entfernt, wenn sie nicht befahren werden kann oder darf, zum Beispiel durch eine verringerte Fahrbahnbreite aufgrund von Baustellen.

In Kombination mit zusätzlichen Informationen über das Fahrzeug des Servicetechnikers, sprich Höhe, Breite, Länge und Gewicht, da auch große Ersatzteile transportiert werden, lässt sich eine individuelle Routenplanung durchführen. Das Ergebnis einer solchen Berechnung ist in der folgenden Grafik dargestellt.



### SmartCharging-Use-Case

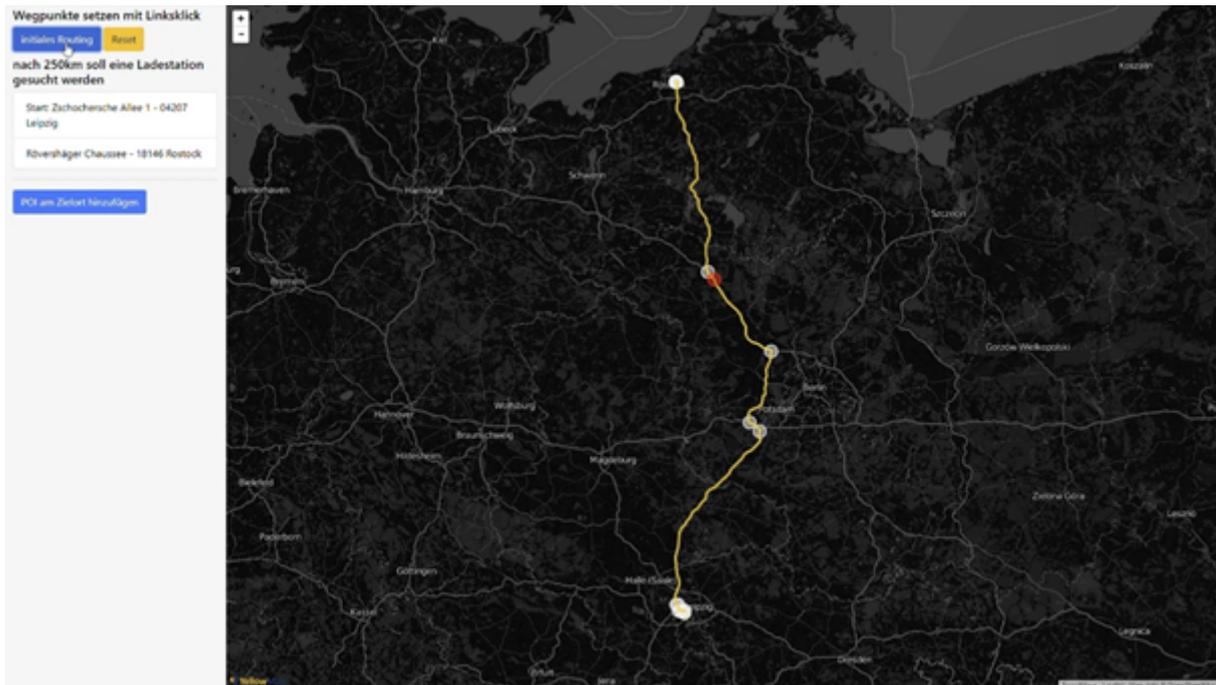
Der Anwendungsfall Smart Charging lehnt sich an die Produktentwicklung der YellowMap im Bereich elektromobiles Laden an. Die YellowMap verfügte initial über ein qualitätsgesichertes Verzeichnis von Ladestationen, welches alle wichtigen Parameter für den Fahrer umfasst. So sind bspw. Betreiber, Abrechnungsmodalitäten und damit Zugänglichkeit, die unterstützten Steckertypen, ggf. Öffnungs-

respektive Zugangszeiten, sowie Bilder zur besseren Erkennung aus dem fahrenden Fahrzeug verfügbar. Eine Live-Datenanbindung (Belegungszustand der Stationen) wurde in Zusammenarbeit mit den Betreibern umgesetzt. Da sich die Belegung zwischen Fahrtantritt und Erreichen der Ladestation ändern kann, wurde zudem eine statistische Ladezustandsprognose entwickelt, um dem Fahrer mehr Sicherheit bei seiner Planung zu geben.

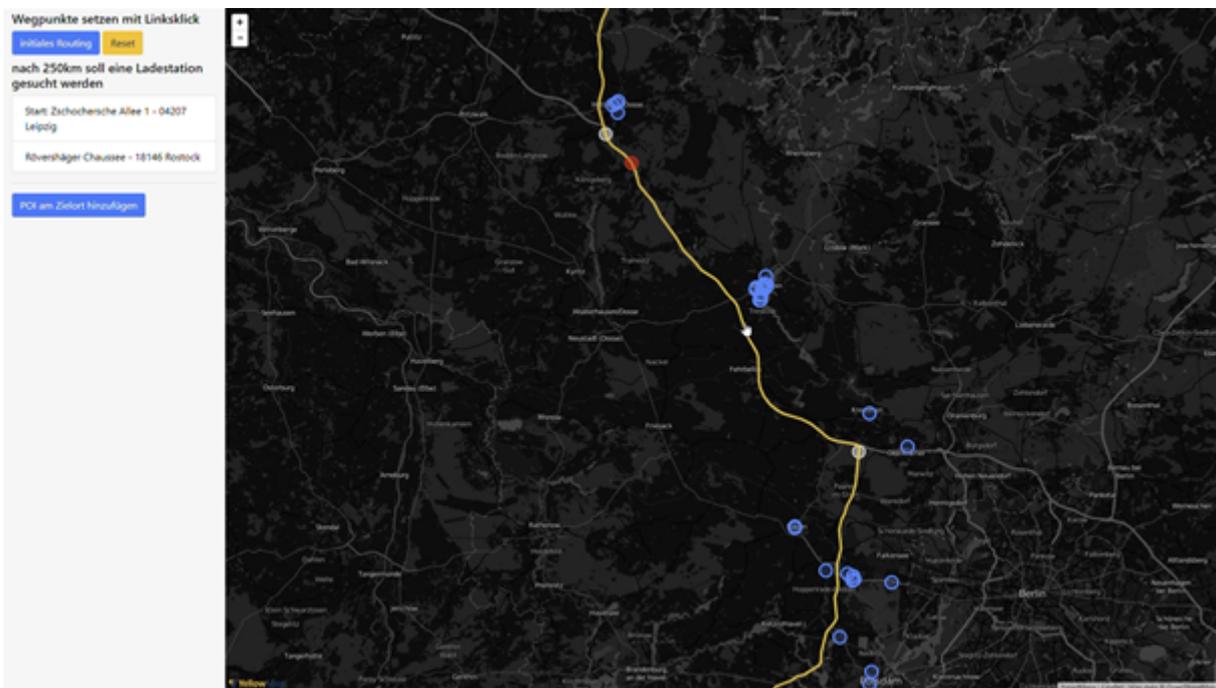
Im Limbo-Projekt sollte dieser Service einerseits um weitere nützliche Informationen zur Lage der Station bezüglich Erreichbarkeit eines Zieles, sowie nahegelegener POI und um eine Reiseplanung erweitert werden, die ihn unterstützt, wenn er sein Ziel nur durch (mehrfaches) Laden unterwegs erreichen kann. Diese Reise- und integrierte Ladeplanung ist kein klassisches Routing, denn die Zwischenladeintervalle lassen sich in klassischen Shortest-Path-Algorithmen nicht integrieren. Auch viele Optimierungen, die solche Algorithmen verwenden, sind im SmartCharging-Use-Case oft nicht anwendbar. Hinzu kommt, dass es aus Fahrersicht ggf. sinnvoller sein kann, Ladepausen mit anderen Aktivitäten zu verbinden und so die Zeit sinnvoll zu nutzen. Solche Optionen hängen davon ab, wo sich eine angefahrene Ladestation befindet, welches Angebot erreichbar ist und welche Aktivitäten der Fahrer als nützlich empfindet. Der Nutzen lässt sich jedoch kaum generalisieren, daher sollte die LIMBO-Lösung die Ladestationen nach möglichst frei wählbaren Kriterien bewerten. Gerade hierfür ist eine semantische Abfrage hervorragend geeignet. Im Bereich e-Mobility wurde hierzu ein Demonstrator umgesetzt, der eine Routenplanung (eigentlich Reiseplanung) unter Verwendung einer flexiblen semantischen Abfrage über geliftete Stationsdaten ermöglicht. Das Ergebnis ist daher auch kein fixer Vorschlag, sondern eine grafisch aufbereitete Auswahl. Die Entscheidung trifft der Fahrer selbst. Der Demonstrator wurde anhand von drei Beispielszenarien mit wachsender Komplexität entwickelt:

- Das erste Beispiel bestand in einem Anreise-Abreise-Szenario mit Ladebedarf erst am Zielort (Zielort liegt innerhalb der Reichweite des Fahrzeugs, aber für den Rückweg ist Laden erforderlich). Hierzu müssen Lademöglichkeiten selektiert werden, die einer Reihe verschiedener Kriterien genügen. Die Ladestation sollte zum Fahrzeug passen, die/eine passende Ladekarte erfordern, zur Ankunftszeit voraussichtlich verfügbar sein und - entweder im Nahbereich des Ziels liegen oder (hierfür eignet sich LIMBO) mit anderen vor Ort oder im Nahbereich verfügbaren Verkehrsmitteln gut an den Zielort angebunden sein. Zusätzlich sind weitere Kriterien über die Abfrage frei definierbar. In dieser einfachen Form des Demonstrators wird davon ausgegangen, dass das Fahrzeug während des Aufenthalts geladen werden kann, ohne dass es zu weiteren Wartezeiten kommt. Dies trifft beispielsweise auf Pendlerstrecken, einen Besuch oder eine längere Dienstleistung vor Ort zu. Eine Prüfung auf möglichst schnelle Ladung ist daher entbehrlich. Verzögerungen entstehen damit nicht durch das Laden selbst, sondern durch Wartezeiten an der belegten Ladestation sowie durch schlechte(re) Anbindung an die Destination mittels anderer Mobilitätsmodi.
- In zweiten Szenario fährt der Nutzer ein entfernt liegende Ziel an, hat dort aber nur einen kurzen Aufenthalt - bspw. ein kurzer Einkauf, ein Hol/Bring-Fahrt oder Kurierfahrt.. Es gelten dieselben Kriterien wie zuvor, jedoch entfällt die zuvor geltende Vereinfachung, dass der Aufenthalt vor Ort für die Ladung des Fahrzeugs (allein) ausreicht. Daher kann es in diesem Szenario sinnvoll sein, den Ladestopp in weiterer Distanz zum Ziel zu planen oder ggf. auch von der direkten Route abzuweichen, wenn sich dadurch (Verfügbarkeit einer Schnellladung) die Gesamtdauer erheblich verkürzen lässt. Hierbei können weitere Kriterien abgefragt werden - bspw. die Nähe der Ladestation zu POI verschiedener Art (Café, Kinderspielplatz, Einkaufsmöglichkeiten,...). In diesem Szenario ist es daher erforderlich, Ladestationen auf der Route mit zu betrachten, nach entsprechenden Kriterien zu bewerten und dem Nutzer zur Wahl zu stellen.
- Im dritten Szenario mit der höchsten Komplexitätsstufe fährt der Fahrer eine Strecke zum Ziel, die die Reichweite seines Fahrzeugs übersteigt. Er muss also mehrfach Zwischenstopps einlegen, um das Fahrzeug zu laden. Die Ladestationen unterwegs werden ihm nach vorgegebenen Kriterien empfohlen und so ausgewählt, dass er die Reichweite möglichst

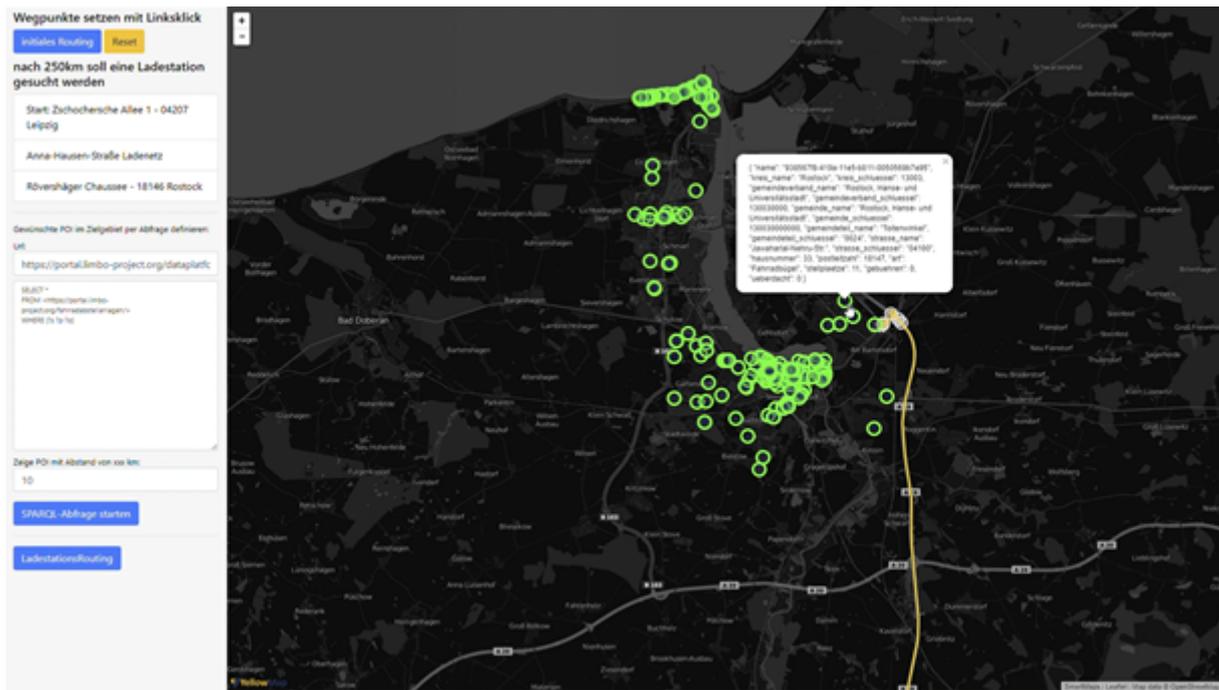
ausnutzt. Die nachfolgenden Screenshots zeigen eine solche Fahrt.



Die Abbildung zeigt die initiale Wahl der Route von Leipzig nach Rostock. Der rote Punkt markiert die Reichweite des Fahrzeugs von etwa 250 km ab Start.



In dieser Abbildung werden die verfügbaren und vorausgewählten Ladestationen entlang der Zielroute eingeblendet.



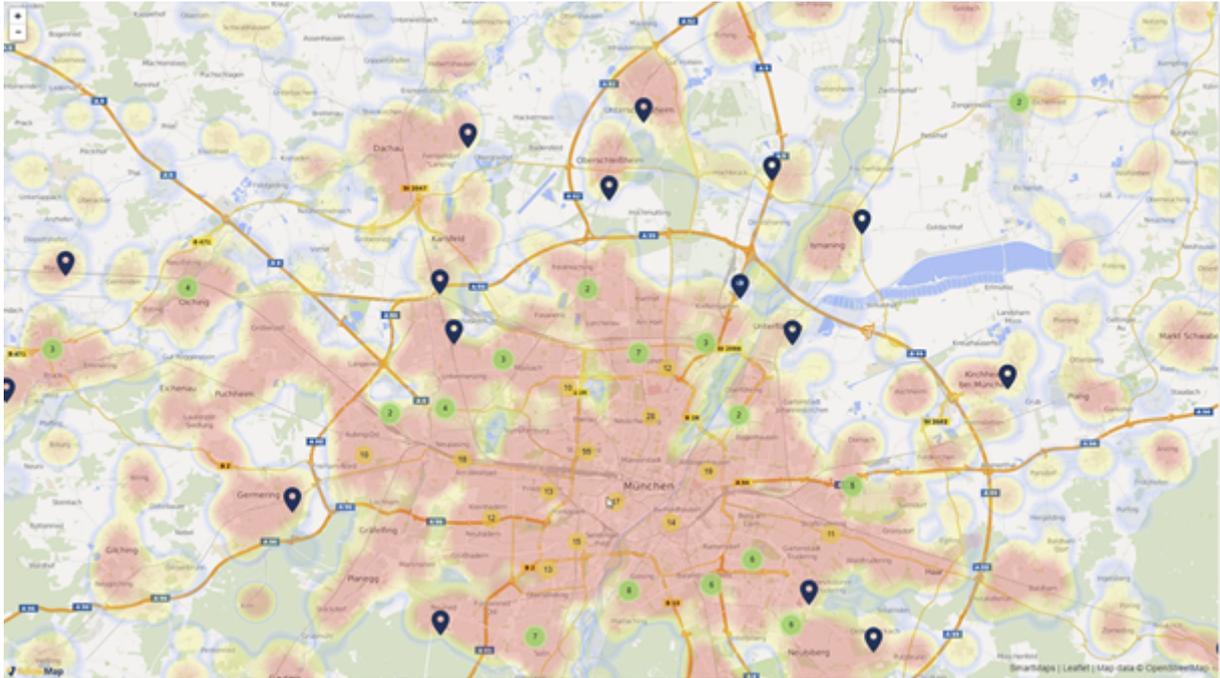
In dieser Abbildung wählt der Fahrer in Rostock angekommen eine Ladestation aus, die entweder im Nahbereich seiner Destination liegt oder von dieser über ein alternatives Verkehrsmittel erreicht werden kann, während das Fahrzeug geladen wird.

### Standortplanung für Infrastrukturen

Die Planung vieler Infrastrukturen setzt je nach Art der Infrastruktur häufig eine Berücksichtigung technischer Eignungsparameter sowie örtlich vorliegender Bedarfe oder auch sozio-demographischen Eignungsparameter für eine gute Planungsentscheidung voraus. Zu beiden Kriterien müssen somit sinnvolle Informationen gesammelt, verarbeitet und aggregiert werden. YellowMap und CISS TDI haben hierbei zwei Szenarien diskutiert und anschließend eine in einer Demonstration umgesetzt.

Das nicht umgesetzte, aber für eine Umsetzung durchaus mögliche Szenario bezog sich auf den Anwendungsfall von Windkraftanlagen. Die wesentlichen technischen Bestimmungsfaktoren sind hier die Lagegüte mit Bezug auf eine Ertragsprognose (Windhöufigkeit) und Kosten des Netzanschlusses sowie die Berücksichtigung einer Anzahl von Ausschlusskriterien wie Schutzgebieten, Nähe zu anderen Infrastrukturen, Einhaltung der Abstandsregeln zu Wohnbebauung im Umfeld und dergleichen. Ein zweiter, immer mehr in den Vordergrund tretender Risikofaktor sind jedoch Widerstände in der Bevölkerung, die in die Gründung von Bürgerinitiativen und Einreichung von Klagen münden. Letztere lassen sich ggf. durch vergangene Ereignisse im Umfeld erfassen. Für Standorte, an denen in der Vergangenheit keine WKA geplant wurden, lassen sich u.U. Milleuvorgleiche der ansässigen Bevölkerung mit solchen Regionen anstellen, in denen in der Vergangenheit häufiger Akzeptanzprobleme zu beobachten waren. Allerdings ließ sich hierfür keine geeignete Datengrundlage aufbauen.

Im zweiten Szenario für Standortplanung hat sich die YellowMap gemeinsam mit CISS TDI auf die Planung weiterer Ladestationen konzentriert. Als Entscheidungsparameter sind hier zum einen die technische Realisierbarkeit als auch ein nicht (hinreichend) abgedeckter Bedarf zu berücksichtigen. Der erste Aspekt wurde durch das Weißflächen-Kartierungswerkzeug von CISS-TDI, der zweite durch die Analyse von Bedarfen und bestehenden Angeboten durch YellowMap realisiert. Als Beispiel wurde die Stadt München analysiert. Die nachfolgend dargestellte Karte zeigt Bedarfscluster als HeatMap. Die entsprechenden Hintergrunddaten wurden aus einer Kombination aus Dichte wichtiger Destinationen (Einkaufs, Bildungs und Freizeitangebote) sowie Daten der Ladezustandsprognose gewonnen.

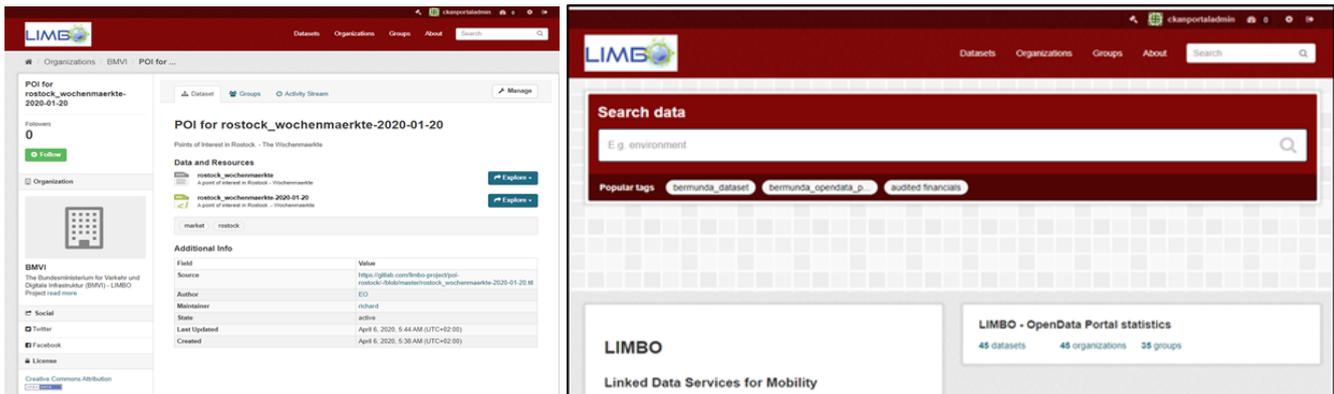


In den grünen und gelben Kreisen sind vorhandene Angebote in den Clustern dargestellt. Wie dort zu sehen, ist im Zentrum Münchens eine relativ hohe Dichte an Ladestationen gegeben. In den peripheren Bereichen sind hingegen Bereiche bisher nicht oder nicht hinreichend abgedeckt. Die blau dargestellten Punkte signalisieren einen Ort, an dem derzeit das Angebot und Nachfrage von Lademöglichkeiten auseinander liegen und welche hernach mittels des Werkzeugs von CISS TDI auf potenzielle Standortkandidaten analysiert wurden.

## Daten und Ontologien

### CKAN-Portal

Daten, die durch das LIMBO-Konsortium erhoben oder veredelt wurden, sind über ein CKAN-Portal zur Verfügung gestellt worden (siehe Abbildung unten). Das Portal erlaubt eine Suche anhand von Stichwörtern und Kategorien und bietet eine CKAN-Schnittstelle an, damit die entsprechenden Metadaten "geharvestet" werden können. Die Harvesting-Funktionalität von CKAN kann anschließend verwendet werden, um Metadaten von anderen Daten-Portalen einzuziehen. Damit kann ein föderiertes Netzwerk von Daten-Portalen geschaffen werden, die untereinander Daten austauschen.



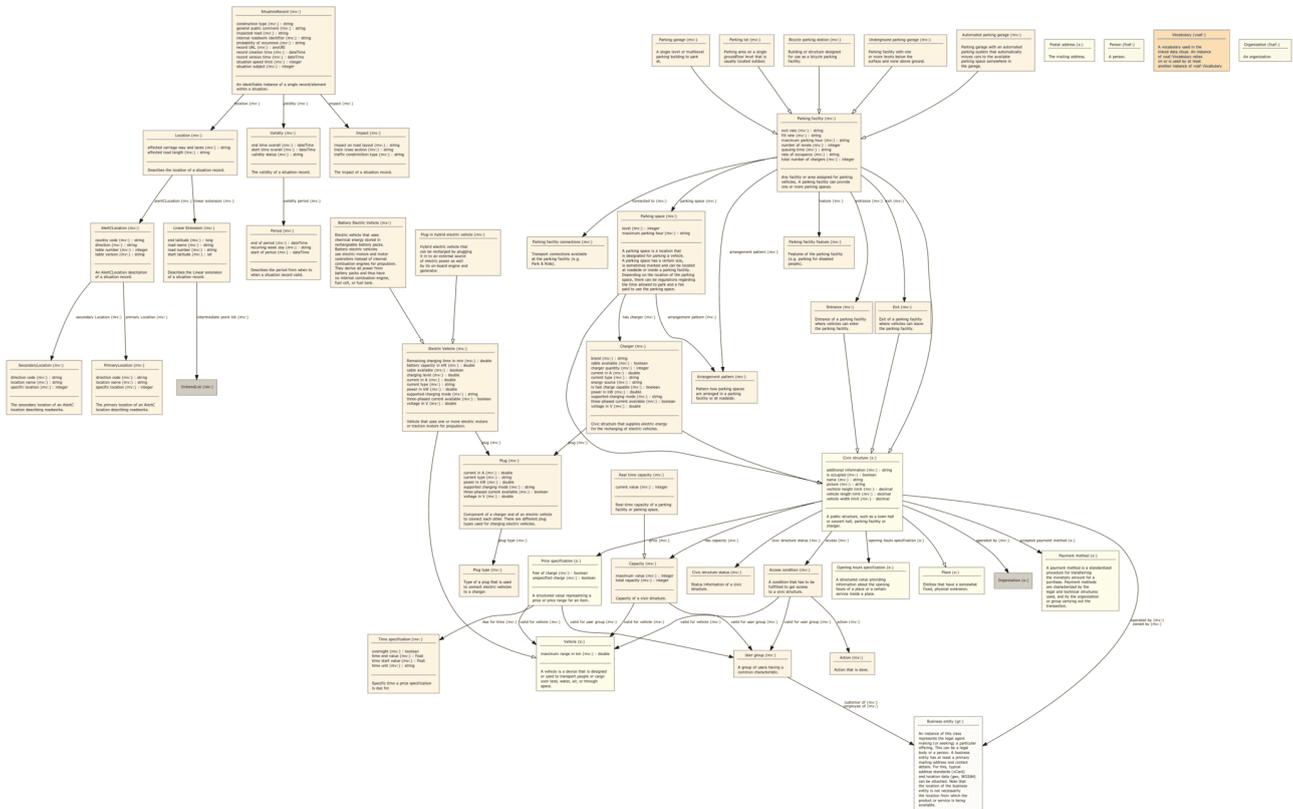
### ALKIS

Die Datenstruktur des Amtlichen Liegenschaftskataster-Informationssystems (ALKIS) verwendet primär ein komplexes XML-Schema, das von weiteren UML-basierten Modellierungen abgeleitet wurde. Hierbei werden primär Geodaten modelliert, die für Landesvermessungsbehörden hohe Relevanz haben. Im Rahmen des LIMBO-Projektes wurde auf Basis von UML-RDF-Ableitungsregeln eine vereinfachte ALKIS-Ontologie<sup>38</sup> entwickelt, die die wesentlichen Bestandteile in Form einer formal definierten semantischen Beschreibung abbildet. Zur Evaluation des Ansatzes wurden ALKIS-Daten aus Hamburg (Flurstücke und Hausumringe) mit Hilfe des RDFTranslators und der Ontologie erfolgreich umgewandelt, veredelt und über einen SPARQL-Endpoint den Partnern zur Verfügung gestellt. Der nächste Screenshot stellt eine Übersicht des Modells dar.

<sup>38</sup> <https://w3id.org/alkis>



Im Rahmen des LIMBO-Projektes wurde von eccenca das offene Mobilitäts-Vokabular MobiVoc<sup>42</sup> weiterentwickelt. Im Ergebnis wurden 4 Releases veröffentlicht, welche u.a. Erweiterungen zur Beschreibung von Parkraum der Straßenarbeiten beinhaltet. Version 1.1.4 vom 22.08.2019 wird im folgenden Screenshot strukturell dargestellt<sup>43</sup>.



### Extraktion aus unstrukturierten Daten

Die Extraktion von strukturiertem Wissen aus unstrukturierten Datenquellen bildete den Kern der Arbeit der Universität Paderborn in AP 4 und besteht im Wesentlichen aus zwei Schritten: der Identifizierung von Entitäten (engl. Named Entity Recognition, kurz NER), der Disambiguierung der gefundenen Entitäten (engl. Named Entity Disambiguation, kurz NED).

Geplant war zunächst die Nutzung des Frameworks FOX,<sup>44</sup> welches mehrere verschiedene NER-Frameworks wie *Stanford NLP* oder *DBpedia Spotlight* nutzt. Da aber zu wenige NLP-Tools ein ausreichendes F1-Maß bei der Erkennung von Entitäten in Mobilitätsdaten erreichen, war der Einsatz von FOX nicht durchführbar. Stattdessen wurde ein auf Rasa-NLU<sup>45</sup> basierendes NER-Modell auf einem händisch generierten Trainingsdatensatz mit insgesamt 277 Entitäten, insbesondere Straßennamen, trainiert. Gleichzeitig wurde das Entity-Linking-Framework AGDISTIS/MAG<sup>46</sup> erweitert, indem die im Lifting-Portal transformierten Daten in einem Suchindex indexiert wurden. Die durch das NER-Modell gefundenen Entitäten dienen als Eingabe von AGDISTIS/MAG.

<sup>42</sup> <https://www.mobivoc.org/>

<sup>43</sup> <https://www.mobivoc.org/en/news/2019/mobivoc-release-en.html>

<sup>44</sup> <https://github.com/dice-group/fox>

<sup>45</sup> <https://rasa.com/>

<sup>46</sup> <http://aksw.org/Projects/AGDISTIS.html>

Zudem wurde der gesamte Extraktionsprozess mithilfe des Frameworks GERBIL<sup>47</sup> kontinuierlich auf einem Benchmark-Datensatz evaluiert. Die Extraktion erreichte ein F1-Maß von 0.67.

## Linking

Das Linking von Mobilitätsdaten umfasste die Entwicklung von Metriken sowie die Konzeptionierung und Evaluierung zeiteffizienter Algorithmen zur Berechnung dieser Metriken. Als Metriken für Mobilitätsdaten sind insbesondere Entfernungen, topologische Beziehungen sowie Überlappungen von räumlichen Objekten von zentraler Bedeutung. Folglich wurden effiziente Algorithmen zur Berechnung von Entfernungen und topologischen Beziehungen entwickelt und in das Linking-Framework LIMES<sup>48</sup> integriert.

### Zeiteffiziente verteilte Berechnung von geo-räumlichen Entfernungen

Für die verteilte Berechnung von georäumlichen Entfernungen implementierten wir den ORCHID-Algorithmus mithilfe der beiden größten Big-Data-Frameworks Apache Spark<sup>49</sup> und Apache Flink<sup>50</sup>. Sowohl Spark als auch Flink arbeiten mit dem verteilten Dateisystem Hadoop<sup>51</sup> und erlauben eine automatische horizontale Skalierung beliebiger Berechnungen, setzen dabei aber jeweils auf andere Techniken und Paradigmen. So werden Berechnungen bei Spark stets auf Batches konstanter Größe angewandt, was einen hohen Datendurchsatz zu Lasten höherer Latenz bedeutet. Flink implementiert Echtzeit-Streaming und kann dadurch eine sehr niedrige Latenz auch bei hohem Durchsatz gewährleisten.

### Zeiteffiziente Berechnung geo-räumlicher topologischer Relationen

Die topologische Relation zweier geo-räumlicher Objekte beschreibt, wie sich die Objekte zueinander verhalten, zum Beispiel ob sie sich überlagern, berühren oder ob das eine Objekt komplett im anderen beinhaltet ist. Diese Relationen können mithilfe des Dimensionally Extended Nine-Intersection Models (DE-9IM) beschrieben werden. Jede Relation wird in diesem Modell mit einer 3x3 Matrix kodiert. Wir benutzen die Eigenschaft dieser Matrizen, mindestens einen Eintrag zu besitzen, der verschieden von Null ist, als Grundlage für unseren Algorithmus Radon.

Da für jede mögliche Relation beide Geometrien mindestens einen Punkt teilen, wenden wir sogenanntes Space-Tiling an, um eine effiziente Indizierung aller Geometrien zu ermöglichen. Dabei wird ein engmaschiges zweidimensionales Raster über den Globus gelegt, und für jede Kachel sammeln wir alle Geometrien, die mindestens einen Punkt auf dieser Kachel besitzen. Nun vergleichen wir nur noch Geometrien, welche in mindestens einer Kachel übereinstimmen. Darüber hinaus wenden wir eine weitere Verbesserung für Relationen an, bei denen eine Geometrie eine strikte Teilmenge einer anderen ist. Hierzu werden *Minimal Bounding Boxes* (MBB) für jede Geometrie berechnet und bevor eine zeitlich aufwendige Berechnung der Schnittmenge der beiden Ursprungsgeometrien erfolgt, überprüfen wir, ob ihre MBBs die strikte Teilmengebeziehung erfüllen. Falls nicht, können wir den Vergleich erneut sparen. Radon2 verbessert die Zeiteffizienz von Radon weiter, indem wir alle möglichen Relationen der DE-9IM-Matrix durch eine Buffering-Technik gleichzeitig berechnen können. Anwendungsfälle, in denen man an mehreren potentiellen Relationen zur Verknüpfung von Mobilitätsdaten interessiert ist, profitieren davon besonders.

---

<sup>47</sup> <http://aksw.org/Projects/GERBIL.html>

<sup>48</sup> <https://github.com/dice-group/limes>

<sup>49</sup> <https://spark.apache.org>

<sup>50</sup> <https://flink.apache.org>

<sup>51</sup> <https://hadoop.apache.org>

## Fusion

Die Fusion von bereits verknüpften Mobilitätsdaten ermöglicht es, eine gemeinsame, kohärente Datenbasis aus einer Vielzahl von Datenquellen zusammenzustellen. Dies ist für viele Anwendungsszenarien wünschenswert. Eine besondere Herausforderung stellt dabei jedoch die konkrete Auswahl und Konfiguration der Fusionsmechanismen dar.

Wir erarbeiteten im Rahmen des LIMBO-Projektes daher eine Lösung zur automatischen Auswahl und Konfiguration von Fusionsmechanismen als Erweiterung des *RDF Dataset Enrichment Frameworks* Deer<sup>52</sup>. Unsere Lösung Adagio nutzt Genetische Algorithmen und basiert auf der Multi-Expression Programming Methode (kurz MEP). Durch Supervised One-Shot Learning müssen Anwender von Deer nicht mehr zwingend Experten für Datenintegration sein. Stattdessen reicht es, ein minimales Beispiel anhand der bestehenden Daten bereitzustellen, um daraus automatisch eine Konfiguration zu lernen. Die gelernte Konfiguration kann danach für die kontinuierliche Integration und Fusion von Mobilitätsdaten benutzt werden.

---

<sup>52</sup> <https://github.com/dice-group/deer>

## Vernetzung

Ein wesentlicher Zweck des mFUND Förderprogramms ist die Vernetzung von Akteuren aus verschiedenen Bereichen, um so Synergien zu erzeugen und dadurch zusätzliche Mehrwerte zu schaffen. Der folgende Abschnitt listet die wichtigsten Vernetzungsaktivitäten auf, bei denen das LIMBO Konsortium beteiligt war und die im Rahmen des Projektes LIMBO durchgeführt wurden.

- **Behindertenverband Leipzig e.V.:** Kooperationspartner und Daten-Lieferant für die BlindSpot-App.
- **Blinden- und Sehbehindertenverband Sachsen e.V. Leipzig:** Kooperationspartner der appPlant GmbH für Nutzer-Tests der BlindSpot-App.
- **Deutsches Zentrum für barrierefreies Lesen (dzb lesen):** Kooperationspartner der appPlant GmbH für die Weiterentwicklung der Indoor-Orientierung für Sehgeschädigte im Folgeprojekt zu ALeite/BlindSpot.
- **Gräbert GmbH, Berlin:** Projektpartner der appPlant GmbH für die Weiterentwicklung der Indoor-Orientierung für Sehgeschädigte im Folgeprojekt zu ALeite/BlindSpot.
- **MDV:** Das InfAI hat im Jahr 2018 ein Treffen mit dem Mitteldeutschen Verkehrsverbund (MDV) zum Thema "Vernetzung von Umwelt- und Verkehrsdaten" organisiert sowie an einer Beiratssitzung des Fachinformationsdienstes "Mobilitäts- und Verkehrsforschung" (FID Move) teilgenommen. Im Rahmen des letztgenannten Treffens wurden v.a. mobilitätsrelevante Aspekte für den Aufbau der Nationalen Forschungsdateninfrastruktur (NFDI) diskutiert.
- **Landkreis Ahrweiler:** Kooperation mit der CISS TDI GmbH dem Ziel, gemeinsam in einem GIS-basierten Prozess Potentialflächen für Windkraftanlagen im Kreis Ahrweiler zu identifizieren.
- **LSWT:** Weitere Vernetzungsaktivitäten haben auf dem 7. Leipziger Semantic Web Tag (LSWT) im Jahr 2019 stattgefunden. Bei dieser von der eccenca und dem InfAI organisierten Veranstaltung wurde das LIMBO-Projekt einem Publikum aus Vertretern von Industrie und Wissenschaft vorgestellt.
- **mFUND-Arbeitsforen:** Außerdem haben Projektmitarbeiter an den mFUND-Arbeitsforen zu den Themen *Datenbankarchitekturen* und *Mobilität in ländlichen Räumen* teilgenommen.
- **OPAL<sup>53</sup>:** Opal ist ein mFUND Projekt, welches von der Universität Paderborn durchgeführt wurde. Ziel des Projekts war die Entwicklung eines ganzheitlichen, offenen Datenportals. Insbesondere wurde im Bereich der Datenextraktion und der Suche während der gesamten Projektlaufzeit eng mit dem OPAL Projekt zusammengearbeitet. Zudem fand ein regelmäßiger Austausch statt.
- **Universität Leipzig, Institut für Informatik, BIS:** Projektpartner der appPlant GmbH für die Weiterentwicklung der Indoor-Orientierung für Sehgeschädigte im Folgeprojekt zu ALeite/BlindSpot.
- **Workshop "Mit Open Data in die Zukunft":** Gemeinsam mit dem WIK veranstaltete die Firma CISS TDI GmbH einen offenen Workshop, der sich an Forschungseinrichtungen und Unternehmen richtete, die sich vermehrt mit der Thematik Open Data befassen.

---

<sup>53</sup> <http://projekt-opal.de/>

## Publikationen

Nachfolgend sind die im Rahmen des Projektes entstandenen Publikationen in chronologisch absteigender Reihenfolge aufgeführt.

### 2020

- Figura, R. (2020). Efficient data collection with heterogeneous cooperating objects for environmental monitoring. Doctoral dissertation, University of Duisburg-Essen, Germany.
- Georgala, K., Röder, M., Sherif, M. A. et al. (2020). Applying edge-counting semantic similarities to Link Discovery: Scalability and Accuracy. Proceedings of Ontology Matching Workshop 2020
- Georgala, K., Sherif, M. A. & Ngonga Ngomo, A.-C. (2020). LIGER – Link Discovery with Partial Recall. Proceedings of Ontology Matching Workshop 2020.
- Neubacher, A., Kraft, A., Willner, A. et al. (2020). Vorschlag zur systematischen Klassifikation von Interaktionen in Industrie 4.0 Systemen.
- Obinna, E. & Figura, R. (2020). CKAN-WIT - Ein Werkzeug für Metadaten-Aggregation, -Filterung und -Präsentation.
- Sherif, M. A., Kevin Dreßler & Ngonga Ngomo, A.-C. (2020). LIGON – Link Discovery with Noisy Oracles. Proceedings of Ontology Matching Workshop
- Schröder, T., Willner, A. & Diedrich, C. (2020). Semantische Interoperabilität in Cyber-physischen Produktionssystemen. In B. Ege (ed.), Semantische Datenintelligenz im Einsatz
- Stadler, C., Bin, S., Wenige, L., Bühmann, L., & Lehmann, J. (2020). Schema-agnostic SPARQL-driven faceted search benchmark generation. Journal of Web Semantics, 65.
- Wenige, L., Stadler, C., Bin, S. et al. (2020). Automatic Subject Indexing with Knowledge Graphs. LASCAR Workshop at the Extended Semantic Web Conference (ESWC).
- Zahera, H. M. & Sherif, M. A. (2020). ProBERT: Product Data Classification with Fine-tuning BERT Model. Proceedings of Mining the Web of HTML-embedded Product Data Workshop (MWP2020)

### 2019

- Ahmed, A. F., Sherif, M. A. & Ngonga Ngomo, A.-C. (2019). Do your Resources Sound Similar? On the Impact of Using Phonetic Similarity in Link Discovery. K-CAP 2019: Knowledge Capture Conference
- Ahmed, Abdullah Fathi., Sherif, Mohamed Ahmed. & Ngonga Ngomo, A.-C. (2019). LSVS: Link Specification Verbalization and Summarization. 24th International Conference on Applications of Natural Language to Information Systems (NLDB 2019), : Springer.
- Arndt, N. & Martin, M. (2019). Decentralized Collaborative Knowledge Management using Git (Extended Abstract). Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May, San Francisco, CA, USA.
- Arndt, N. & Radtke, N. (2019). Conflict Detection, Avoidance, and Resolution in a Non-Linear RDF Version Control System: The Quit Editor Interface Concurrency Control. Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May, San Francisco, CA, USA.
- Arndt, N., Zänker, S., Sejdiu, G. et al. (2019). Jekyll RDF: Template-Based Linked Data Publication with Minimized Effort and Maximum Scalability. 19th International Conference on Web Engineering (ICWE 2019), June, Daejeon, Korea.
- Desouki, A. A., Röder, M. & Ngonga Ngomo, A.-C. (2019). Ranking on Very Large Knowledge Graphs. Proceedings of the 30th ACM Conference on Hypertext and Social Media (p./pp.

163--171).

- Figura, R., Willner, A. & Martin, M. (2019). Smarte Daten im Knowledge Graph, die Grundlage einer zukunftssicheren Bereitstellung Offener Daten
- Gusmita, R. H., Jalota, R., Vollmers, D. et al. (2019). QUANT - Question Answering Benchmark Curator.. In M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack & Y. Sure-Vetter (eds.), SEMANTICS (p./pp. 343-358), : Springer.
- Kaur, P., Blücher, V., Jalota, R., Moussallem, D., Ngomo, A.-C. N. & Usbeck, R. (2019). Offline Question Answering over Linked Data using Limited Resources.. In M. Alam, R. Usbeck, T. Pellegrini, H. Sack & Y. Sure-Vetter (eds.), SEMANTICS Posters&Demos, : CEUR-WS.org.
- Moussallem, D., Soru, T. & Ngomo, A.-C. N. (2019). THOTH: Neural Translation and Enrichment of Knowledge Graphs.. In C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. F. Cruz, A. Hogan, J. Song, M. Lefrançois & F. Gandon (eds.), ISWC (1) (p./pp. 505-522).
- Obinna, E. & Figura, R. (2019). Building Geo-Spatial Data Portal for optimal User experience with CKAN Core-bundle and OPENLAYER Library.
- Obraczka, D. & Ngonga Ngomo, A.-C. (2019). Dragon: Decision Tree Learning for Link Discovery.. In M. Bakaev, F. Frasincar & I.-Y. Ko (eds.), ICWE (p./pp. 441–456), : Springer.
- Stadler, C., Wenige, L., Tramp, S., Junghanns, K. & Martin, M. (2019). RDF-based Deployment Pipelining for Efficient Dataset Release Management. Proceedings of the Posters and Demos Track of the 14th International Conference on Semantic Systems co-located with the 14th International Conference on Semantic Systems (SEMANTICS'19), September, Karlsruhe, Germany.
- Syed, Z. H., Röder, M. & Ngomo, A.-C. N. (2019). Unsupervised Discovery of Corroborative Paths for Fact Validation.. In C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. F. Cruz, A. Hogan, J. Song, M. Lefrançois & F. Gandon (eds.), ISWC (1) (p./pp. 630-646), : Springer.
- Zahera, H. M., Elgendy, I. A., Jalota, R. & Sherif, M. A. (2019). Fine-tuned BERT Model for Multi-Label Tweets Classification. Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019
- Zahera, H. M., Sherif, M. A. & Ngonga Ngomo, A.-C. (2019). Jointly Learning from Social Media and Environmental Data for Typhoon Intensity Prediction. K-CAP 2019: Knowledge Capture Conference (p./pp. 4)

## 2018

- Athreya, R. G., Ngomo, A. N. & Usbeck, R. (2018). Enhancing Community Interactions with Data-Driven Chatbots-The DBpedia Chatbot. Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018 (p./pp. 143–146).
- Choi, K., Anke, L. E., Declerck et al. (2018). Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA.
- Figura, R., Willner, A. & Martin, M. (2018). Applying Linked Data Paradigms for Reagional Weather Data Reanalysis . In I. S. on Regional Reanalysis (ISSR) 2018 (ed.).
- Figura, R. & Willner, A. (2018). Wertschöpfung durch Linked (Open) Data in Unternehmensprozessen.
- Moussallem, D., Usbeck, R., Röder, M. et al. (2018). Entity Linking in 40 Languages using MAG. The Semantic Web, ESWC 2018, Lecture Notes in Computer Science.
- Usbeck, R., Ngomo, A. N., Conrads, F. (2018). 8th Challenge on Question Answering over Linked Data (QALD-8). Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and

9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, USA, (p./pp. 51--57).

- Usbeck, R., Gusmita, R. H., Ngomo, A. N. (2018). 9th Challenge on Question Answering over Linked Data (QALD-9) (invited paper). Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, USA (p./pp. 58--64).
- Zahera, H. M., Jalota, R. & Usbeck, R. (2018). DICE @ TREC-IS 2018: Combining Knowledge Graphs and Deep Learning to Identify Crisis-Relevant Tweets. Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018, Gaithersburg, Maryland, USA.

## 2017

- Figura, R. & Willner, A. (2017). Das Linked Open Data Warehouse.
- Moussallem, D., Usbeck, R., Röder, M. et al. (2017). MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach. K-CAP 2017: Knowledge Capture Conference (p./pp. 8).
- Ngonga Ngomo, A.-C., Hoffmann, M., Usbeck, R. & Jha, K. (2017). Holistic and Scalable Ranking of RDF Data. 2017 IEEE International Conference on Big Data (p./pp. 10).
- Petersen, N., Similea, A., Lange, C. (2017). TurtleEditor: A Web-based RDF Editor to Support Distributed Ontology Development on Repository Hosting Platforms. International Journal on Semantic Computing, 11, 311--323.
- Saleem, M., Dastjerdi, S. N., Usbeck, R. et al. (2017). Question Answering Over Linked Data: What is Difficult to Answer? What affects the F scores?. Natural Language Interfaces workshop at ISWC.
- Usbeck, R., Hoffmann, M., Röder, M. et al. (2017). Using Multi-Label Classification for Improved Question Answering. CoRR, abs/1710.08634.
- Usbeck, R., Ngomo, A. N., Kim, J. (2017). Joint Proceedings of BLINK2017: 2nd International Workshop on Benchmarking Linked Data and NLIWoD3: Natural Language Interfaces for the Web of Data co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria.