TEXTE
79/2023

Abschlussbericht

Umsetzungskonzept umwelt.info

Anhang A: Architektur Dokumentation

von:
Dr. Matthias Bluhm, Dr. Udo Einspanier, Dr. Thore Fechner, Rico Illes, Dr. Matthias Seuter, Dr. Uwe Voges
con terra GmbH, Münster

Herausgeber:
Umweltbundesamt

TEXTE 79/2023

Ressortforschungsplan des Bundesministeriums für
Umwelt, Naturschutz und nukleare Sicherheit

Forschungskennzahl 3720 12 101 0
FB001071

Abschlussbericht

Umsetzungskonzept umwelt.info

Anhang A: Architektur Dokumentation

von

Dr. Matthias Bluhm, Dr. Udo Einspanier, Dr. Thore
Fechner, Rico Illes, Dr. Matthias Seuter, Dr. Uwe Voges
con terra GmbH, Münster

Im Auftrag des Umweltbundesamtes

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Abkürzung	Beschreibung
API	Application Programming Interface
CAS	Central Authentication Service
CMS	Content Management System
CPU	Central Processing Unit
CRUD	Create/Read/Update/Delete
CUD	Create/Update/Delete
CS	Catalogue Service
CSW	Catalogue Service Web
DB	Datenbank
DCAT	Data Catalogue Vocabulary
DCAT-AP	Data Catalogue Vocabulary-Application Profile for data portals in Europe
DS	Daten Service
DU	Unstrukturierte Daten
ETL	Extract Transform Load
GSB	Government Site Builder
ID	Identification
INSPIRE	Infrastructure for Spatial Information in the European Community
ISO	Internationale Organisation für Normung
ITZBund	Informationstechnikzentrum Bund
KI	Künstliche Intelligenz
LD	Linked Data
LDAP	Lightweight Directory Access Protocol
MD	Metadaten
MQA	Metadata Quality Assessment
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
OGC	Open Geospatial Consortium
RAM	Random-Access Memory
RDF	Resource Description Framework
REST	Representational State Transfer
RS	Recommender System
SDW	Spatial Data on the Web
SSO	Single Sign-On
UCD	User-Centered Design
UI	User Interface
URI	Uniform Ressource Identifier
URL	Uniform Ressource Locator

Abkürzung	Beschreibung
VM	Virtuelle Maschine

1 Einführung

Methodisch und inhaltlich basiert die vorliegende Systemarchitektur auf dem Arc42 Architektur Template [1], das eine Vorlage zur Entwicklung, Dokumentation und Kommunikation von Software- und Systemarchitekturen bietet. Die Themen „Technische Randbedingungen und Qualitätsziele“, „Kontextabgrenzung“ und „Lösungsstrategie“ sind in Kap. 3 des Umsetzungskonzepts enthalten, zu dem dieses Dokument einen Anhang darstellt (Anhang A). Dieser Anhang dient der detaillierten Beschreibung der Systemarchitektur und enthält die Bausteinsicht mit der Zerlegung des Systems in Komponenten, die Laufzeitsicht, die Verteilungssicht sowie querschnittliche Konzepte und Entwurfsentscheidungen. Die genannten Kapitel des Hauptdokuments in Verbindung mit diesem Dokument stellen eine IT-Konzeption als Basis für die anschließende Umsetzung dar. Die Systemarchitektur basiert auf den Ergebnissen der Machbarkeitsstudie [2] und den durchgeführten Workshops zu den Themen „Linked Data“ und „Künstliche Intelligenz“ (KI).

Die wesentlichen Zielgruppen für die Systemarchitektur umfassen die Personen, die auf Basis der Vorgaben der Systemarchitektur an umwelt.info arbeiten oder in Entscheidungen über das System und dessen Entwicklung beteiligt sind. Dies sind insbesondere das Entwicklungsteam, die Projektleitung im Umweltbundesamt sowie die betreibende Stelle.

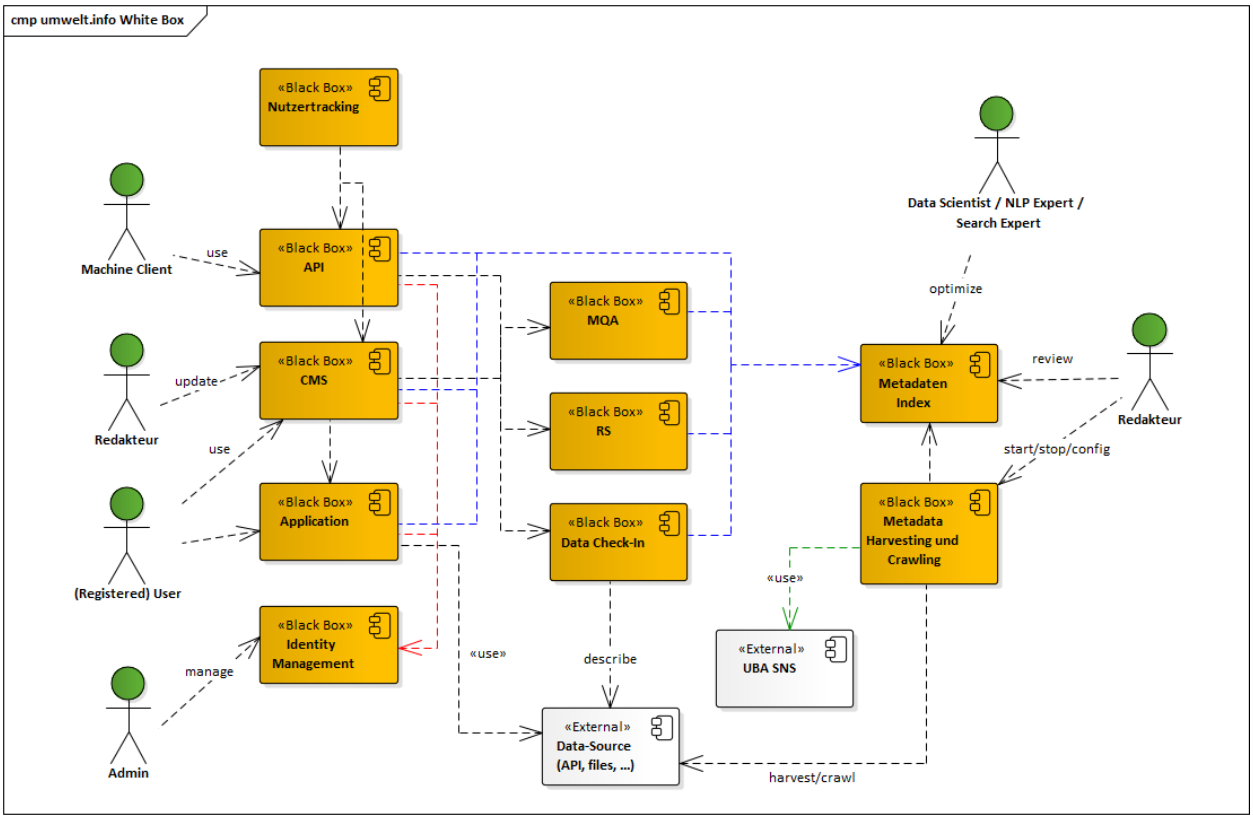
2 Bausteinsicht

In diesem Kapitel erfolgt eine statische Zerlegung des umwelt.info Portals (vgl. Kapitel 3.4 im Umsetzungskonzept) in Komponenten sowie deren Beziehungen. Diese Bausteinsicht bildet den Grundrissplan für die Architektur von umwelt.info. Konkret besteht dieser Plan aus einer hierarchischen Sammlung von sog. Black Boxes und White Boxes und deren Beschreibungen. Während bei einer Black Box nur das äußere Verhalten der Komponente betrachtet wird, wird bei einer White Box auch die innere Struktur dargestellt. Eine White Box besteht aus mehreren Black Boxes, welche in der nächsten Ebene wieder in eine White Box mit Black Boxes zerlegt werden. So ist beispielsweise in Abbildung 1 das gesamte umwelt.info-Portal als White Box mit seinen Komponenten (Black Boxes) dargestellt. In den weiteren Ebenen werden die Black Boxes dann mit ihrer inneren Struktur als White Box dargestellt. Abbildung 3 erweitert die erste Ebene um spezielle Linked Data und KI-Komponenten.

2.1 Ebene 1 umwelt.info (White Box)

Die oberste Ebene entspricht der am stärksten generalisierten Sicht auf das System (Abbildung 1) und zeigt das Zusammenspiel der Komponenten, die im Falle von umwelt.info für das Bekanntmachen und Indizieren der Daten und Informationen und die Suche nach verfügbaren Daten und Informationen notwendig sind. Sie berücksichtigt auch die Möglichkeit, dass Nutzende eigene Daten erzeugen und in das System integrieren können, die dann ebenfalls in die Indizierung und Suche einbezogen werden.

Abbildung 1: Ebene 1 umwelt.info (White Box)



Quelle: eigene Darstellung, con terra GmbH

In den folgenden Tabellen werden die Komponenten inklusive ihrer Schnittstellen, Qualitäts- und Leistungsmerkmale, sowie weitere Details und offene Punkte beschrieben.

2.1.1 Content Management System (CMS)

Beschreibung

Der Einstiegspunkt des Portals ist das Content Management System, das eine Benutzerschnittstelle (Website) bereitstellt, über die Nutzer*innen vor allem nach unterschiedlichen Daten zu verschiedenen Themenbereichen suchen können.

Weitere Details

Das CMS speichert die Inhalte der Seiten des umwelt.info Portals und visualisiert diese für die Auslieferung an die Nutzer*innen, wenn diese eine Portalseite mit ihren Webbrowsern anfragen. Die Portalseiten integrieren die User Interface (UI-)Komponenten des umwelt.info Portals, welche spezielle Funktionen über die Benutzerschnittstelle bieten. Dabei sind einige Funktionen (z. B. die Suche) für alle Nutzer*innen verfügbar und andere nur für angemeldete bzw. berechtigte Nutzer*innen (z. B. Personalisierte Listen bzw. Daten bereitstellen).

Schnittstelle(n)

Bereitgestellt:

- Websites (User-Interface Komponenten) des umwelt.info Portals

Benötigt u.a.:

- MD-Search des Metadaten-Index (s. 2.1.4)
- Identity Management

UI-Komponenten zur Einbettung in Websites

Qualitäts-/Leistungsmerkmale

Besonders hohe Verfügbarkeit und Skalierbarkeit, da es sich um den Haupteinstiegspunkt in das System handelt und die Anzahl gleichzeitiger Nutzer a priori unbekannt ist und stark schwanken kann.

2.1.2 Metadata Harvesting und Crawling

Beschreibung

Das Metadata Harvesting bzw. Crawling (genaue Definition der Unterschiede s. Kap. 2.13) ermittelt aus den verschiedenen Datenquellen außerhalb des Systems die Metadaten, indiziert diese und macht sie somit auffindbar

Weitere Details

Eine weitere Aufgabe ist das Überführen („mappen") der ermittelten Metadaten in das gemeinsame im Metadaten-Index verwendete Metadaten Schema (vgl. Kap. 5.1.1) und die Durchführung von Optimierungen wie z. B. Qualitätsverbesserungen.

Schnittstelle(n)

Bereitgestellt:

- Für das Metadata Harvesting bzw. Crawling wird eine Schnittstelle bereitgestellt, so dass das Harvesting bzw. Crawling für einzelne Datenquellen gestartet, gestoppt und konfiguriert werden kann.
- Start/Stop/Config: Dieses ist die Schnittstelle mit der einzelne Metadata-Harvester bzw. Crawler gestartet bzw. gestoppt und konfiguriert werden können.

Benötigt:

- Das Metadata Harvesting benötigt Zugriff auf die Schnittstellen der externen (Meta-)Daten-Ressourcen. Diese Schnittstellen sollten idealerweise eine Möglichkeit besitzen, lediglich die Daten/Metadaten anfordern zu können, die sich seit dem letzten „Harvesting" geändert haben, um nicht immer alle Daten/Metadaten der Daten-Ressource wiederholt harvesten zu müssen., sondern tatsächlich nur die neu hinzu gekommenen bzw. die veränderten.

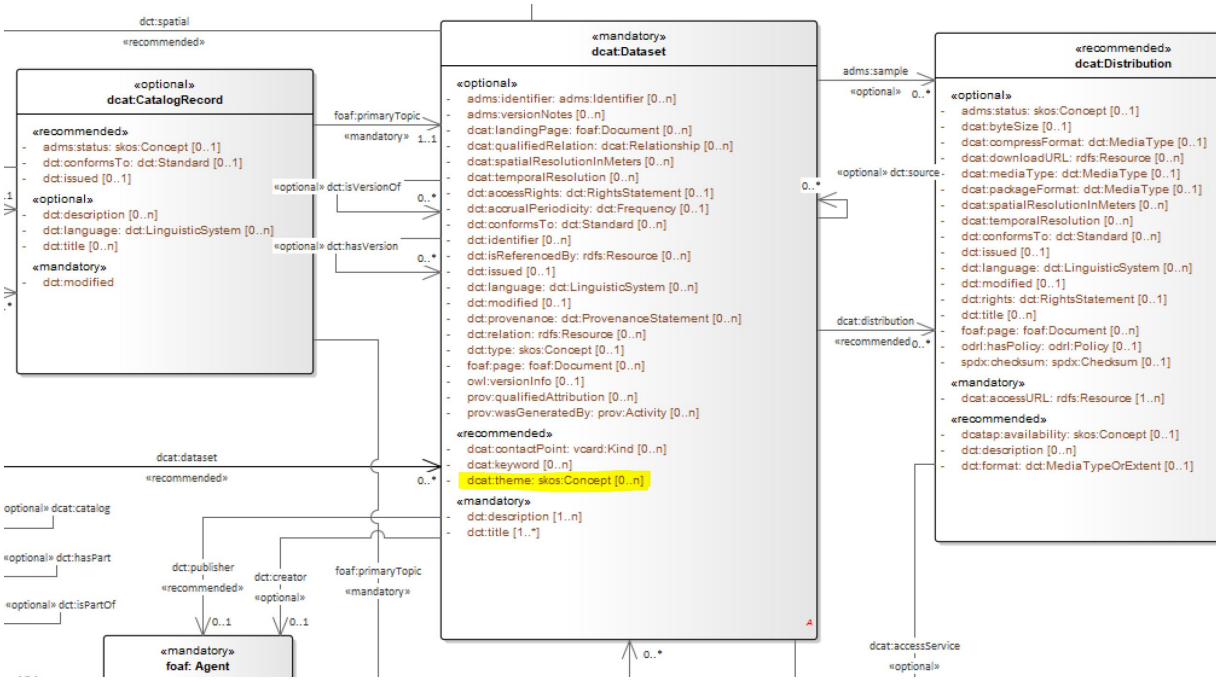
Qualitäts-/Leistungsmerkmale

Das Metadata Harvesting muss verschiedene Datenquellen parallel harvesten können.

Die Adaptoren müssen in der Lage sein, bestimmte Filterungen der (Meta-)Daten durchzuführen, zusammen mit den Transformatoren semantische Tests vorzunehmen und insgesamt die Qualität der abgeleiteten Metadaten zu verbessern.

Eine wichtige Qualitätsverbesserung ist das semantische Annotieren von Metadaten mit Elementen aus einem existierenden Vokabular oder einer Ontologie. Dabei würden im konkreten Fall Links zu Konzepten des Vokabulars entsprechend den dafür vorgesehenen Möglichkeiten des DCAT-AP Metadaten-Modells (s. 5.1.1.1) hinzugefügt (s. gelbe Markierung im u. g. UML-Diagramm).

Abbildung 2: Ausschnitt aus dem DCAT-AP Klassendiagramm



Quelle: DCAT-AP Klassendiagramm: https://github.com/SEMICeu/DCAT-AP/releases/tag/v2.1.1

…

```
<dcat:Dataset rdf:about="https://ckan.open.nrw.de/dataset/DC4501B4-4E9B-4C7E-A961-4DBC429C202B">
  <dct:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2021-09-27T00:00:00</dct:modified>
  <dcat:keyword>FFH</dcat:keyword>
  <dcat:keyword>Regional</dcat:keyword>
  <dcat:theme rdf:resource="http://publications.europa.eu/resource/authority/data-theme/AGRI"/>
  <dcat:keyword>Datenbank</dcat:keyword>
  <dcat:keyword>Informationssystem</dcat:keyword>
  <dcat:keyword>Bundesnaturschutzgesetz</dcat:keyword>
  <dcat:keyword>Landschaft</dcat:keyword>
  <adms:identifier>{DC4501B4-4E9B-4C7E-A961-4DBC429C202B}</adms:identifier>
  <dcatde:contributorID>{http://dcat-ap.de/def/contributors/openNRW}</dcatde:contributorID>
  <dcat:keyword>opendata</dcat:keyword>
  <dcat:keyword>LINFOS</dcat:keyword>
  <dct:publisher>
   <foaf:Organization rdf:about="https://ckan.open.nrw.de/organization/d29dfd38-8c05-44ff-93b8-75e2e0927d03">
    <foaf:name>Geoportal</foaf:name>
   </foaf:Organization>
  </dct:publisher>
  <dcat:keyword>Landschaftsinformation</dcat:keyword>
  <dcat:theme rdf:resource="http://publications.europa.eu/resource/authority/data-theme/GOVE"/>
  <dct:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2008-11-14T00:00:00</dct:issued>
  <dcat:keyword>Lebensräume und Biotope</dcat:keyword>
  <dct:description>Der Karten-Layer RAMSAR zeigt die räumliche Verteilung der Feuchtgebiete von internationaler
Bedeutung nach dem Übereinkommen über Feuchtgebiete, insbesondere als Lebensraum für Wat- und Wasservögel, von
internationaler Bedeutung (Ramsar-Konvention 1971). Deutschland ist Mitglied seit 1976. Ursprünglich hatte die Ramsar-
Konvention den Erhalt und die nachhaltige Nutzung (wise use) von Feuchtgebieten als Lebensraum von Wasservögeln zum
Ziel. In den letzten Jahren haben sich die Konventionsziele erweitert und umfassen nun den ganzheitlichen Schutz von
Feuchtgebieten als bedeutende Ökosysteme zum Erhalt der Biodiversität</dct:description>
  <dcat:keyword>LANUV-Kartenlayer</dcat:keyword>
  <dcat:keyword>Naturschutz</dcat:keyword>
  <dcat:keyword>Nature and Landscape</dcat:keyword>
  <dcat:keyword>Fauna-Flora-Habitat-Richtlinie</dcat:keyword>
  <dct:title>Kartenlayer RAMSAR NRW</dct:title>
  <dcat:distribution>
   <dcat:Distribution rdf:about="https://ckan.open.nrw.de/dataset/DC4501B4-4E9B-4C7E-A961-
4DBC429C202B/resource/3046814c-101b-40f0-8532-a54c683afde4">
    <dct:title>WMS Landschaftsinformationssammlung NRW</dct:title>
    <dcat:accessURL
rdf:resource="https://www.wms.nrw.de/umwelt/linfos?SERVICE=WMS&amp;REQUEST=GetCapabilities"/>
    <dct:format>view</dct:format>
    <dct:license rdf:resource="http://dcat-ap.de/def/licenses/dl-zero-de/2.0"/>
    <dct:description>Der WMS LINFOS NRW umfasst wesentliche Inhalte der Landschaftsinformationssammlung (LINFOS)
NRW wie naturschutzfachliche Grundlagendaten, Alleen und Schutzgebiete, etc..</dct:description>
   </dcat:Distribution>
  </dcat:distribution>
  <dct:accessRights>http://inspire.ec.europa.eu/metadata-
codelist/LimitationsOnPublicAccess/noLimitations</dct:accessRights>
…
  <dct:type>http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset</dct:type>
  <dcat:keyword>inspireidentifiziert</dcat:keyword>
  <dcat:theme rdf:resource="http://publications.europa.eu/resource/authority/data-theme/REGI"/>
  <dcat:keyword>Fauna</dcat:keyword>
  <dcat:contactPoint>
   <vcard:Organization rdf:nodeID="Nc996d14c86f44ffb8f1968a7645f59ba">
    <vcard:hasPostalCode>D-45659</vcard:hasPostalCode>
    <vcard:hasStreetAddress>Postbox 101052,D-45610 Recklinghausen</vcard:hasStreetAddress>
    <vcard:fn>Landesamt für Natur, Umwelt und Verbraucherschutz Nordrhein-Westfalen</vcard:fn>
    <vcard:hasCountryName>Nordrhein-Westfalen</vcard:hasCountryName>
    <vcard:hasEmail rdf:resource="mailto:fachbereich21@lanuv.nrw.de"/>
    <vcard:hasLocality>Recklinghausen</vcard:hasLocality>
   </vcard:Organization>
```

```
  </dcat:contactPoint>
  <dct:spatial>
   <dct:Location rdf:nodeID="Nb7d3e596f82242b3b2d830257a8ca672">
    <locn:geometry rdf:datatype="https://www.iana.org/assignments/media-types/application/vnd.geo+json">{"type":
"Polygon", "coordinates": [[[5.8665085, 52.531437], [9.461479, 52.531437], [9.461479, 50.323895], [5.8665085, 50.323895],
[5.8665085, 52.531437]]]}</locn:geometry>
    <locn:geometry rdf:datatype="http://www.opengis.net/ont/geosparql#wktLiteral">POLYGON ((5.8665 52.5314, 9.4615
52.5314, 9.4615 50.3239, 5.8665 50.3239, 5.8665 52.5314))</locn:geometry>
   </dct:Location>
  </dct:spatial>
  <dcat:keyword>Kataster</dcat:keyword>
  <dcat:keyword>Schutzgebiete</dcat:keyword>
  <dcat:keyword>Raumbezogene Information</dcat:keyword>
  <dct:language rdf:resource="http://publications.europa.eu/resource/authority/language/DEU"/>
  <dct:accrualPeriodicity rdf:resource="http://publications.europa.eu/resource/authority/frequency/CONT"/>
  <dcat:keyword>Flora-Fauna-Habitatrichtlinie</dcat:keyword>
  <dcat:distribution>
   <dcat:Distribution rdf:about="https://ckan.open.nrw.de/dataset/DC4501B4-4E9B-4C7E-A961-
4DBC429C202B/resource/ec18d7d7-914d-4703-ba70-0fd98797253b">
    <dct:title>RAMSAR_EPSG25832_Shape.zip</dct:title>
    <dcat:downloadURL
rdf:resource="https://www.opengeodata.nrw.de/produkte/umwelt_klima/naturschutz/linfos/RAMSAR_EPSG25832_Shape.z
ip"/>
    <dcat:accessURL
rdf:resource="https://www.opengeodata.nrw.de/produkte/umwelt_klima/naturschutz/linfos/RAMSAR_EPSG25832_Shape.z
ip"/>
    <dct:license rdf:resource="http://dcat-ap.de/def/licenses/dl-zero-de/2.0"/>
    <dct:format>Shape</dct:format>
    <foaf:page
rdf:resource="http://www.wms.nrw.de/inspire_umwelt/Inspire_Downloaddienst_Schutzgebiete_NRW?REQUEST=GetCapab
ilities&amp;amp;SERVICE=WFS"/>
    <dct:description>RAMSAR_EPSG25832_Shape.zip</dct:description>
   </dcat:Distribution>
  </dcat:distribution>
  …
 </dcat:Dataset>
</rdf:RDF>
```

Wird dieses Dokument[1][2] an die „automatische Verschlagwortungs-Funktion“ des Semantischen Netzwerk Service (SNS) übergeben, werden u.a. folgende Schlagwörter abgeleitet:

- Ramsar-Konvention (URL: https://sns.uba.de/umthes/_00020240)
- Feuchtgebiet (https://sns.uba.de/umthes/_00009468)
- Feuchtgebietökosystem (https://sns.uba.de/umthes/_00051498)

Diese werden dann folgendermaßen in die DCAT-AP Metadaten integriert (und können später für weitergehende Filterungen, Erklärungen, etc in Anwendungen verwendet werden):

```
  <dcat:theme rdf:resource="https://sns.uba.de/umthes/_00020240"/>
  <dcat:theme rdf:resource="https://sns.uba.de/umthes/_00009468"/>
  <dcat:theme rdf:resource="https://sns.uba.de/umthes/_00051498"/>
```

[1] Zur Effizienzsteigerung würden allerdings (bevor das DCAT-Dokument erzeugt wird) nur die wichtigsten Textpassagen (Titel, Beschreibung, Schlüsselworte …) an den SNS-Service übergeben.

[2] Das gilt auch für Web-Sites und pdf-Dokumente

Es können aber auch einzelne Metadaten Elemente, welche als Text Strings vorliegen (wie z. B. keywords), auf Elemente eines existierenden Vokabulars oder einer Ontologie wie den UMTHES „ge-mappt“ werden (welche die originären Werte am besten semantisch widerspiegeln) und (wie oben geschildert) den DCAT-AP Metadaten hinzugefügt werden.

Eine andere Möglichkeit der Qualitätsverbesserung würde darin bestehen, ein Geo-Thesaurus (Gazetteer) „mapping” zu versuchen, um aus den (Meta-)Daten intelligent räumliche Beschreibungen abzuleiten. Wenn etwa ein Datensatz keinen Raumbezug enthält, könnte ein Thesaurus mittels Begrifflichkeiten angefragt werden, die aus unterschiedlichen Metadatenelementen stammen, um einen „match” zu finden. Wenn der Gazetteer einen eindeutigen „match” findet, weist der Adaptor automatisch den gefundenen Raumbezug zu.

Umgekehrt kann der Adapter in dem Fall, dass die Metadaten einen Raumbezug enthalten, der nicht zu den Metadaten passt, die Metadaten als „eventuell nicht plausibel” vermerken.

Es stellt sich im Kontext der Qualitätsverbesserung auch die Frage, ob die Adaptoren in gewissem Rahmen die Verantwortung übernehmen können, anhand der Metadaten (oder sogar durch einen Zugriff auf die Daten) zu entscheiden, ob die Daten „Open Data“ sind oder nicht und Verweise auf entsprechende Lizenzen generieren können. Minimal sollten aber vorhandene Beschreibungen zu Lizenzen, Zugriffs- und Nutzungsrechten in den Ausgangs-Metadaten auf entsprechende Elemente des gemeinsamen Metadatenmodells „ge-mappt” werden. Dieses muss in den exakten Spezifikationen der Adapter vor der Implementierung definiert werden.

Die Adaptoren müssen Unregelmäßigkeiten beim Zugriff auf die Datenquelle oder beim Transformieren protokollieren (z. B. wenn nicht alle notwendigen Metadaten Elemente abgeleitet werden können oder es Inkonistenzen gibt).

Aufgrund der großen Menge an Metadaten muss bedacht werden, dass ein tägliches vollständiges Harvesting evtl. nicht möglich sein könnte. Deswegen sollte s möglich sein, dass die Adaptoren nur bestimmte Teile der Metadaten anfragen können (z. B. nur Metadaten, die sich in den letzten 24h geändert haben).

Von einem Administrator müssen sich die kleinen Prozessketten konfigurieren und die Adaptoren bis zu einem bestimmten Grad anpassen lassen.

Offene Punkte/Probleme/Risiken

Es besteht das Problem, dass ein und derselbe Datensatz mehrmals geharvestet erden könnte, etwa einmal über einen externen Catalogue-Service und dann noch einmal direktüber den eigentlichen Dienst oder die zugehörigenMetadaten, die in mehreren externen Metadatenquellen vorliegen. Idealerweise sollten diese Metadaten zusammengeführt werden.

2.1.3 Data Check-In

Beschreibung (vgl. [2])

Über den Data Check-In lassen sich durch eine manuelle Bereitstellung von Metadaten beliebige weitere Datenquellen beschreiben, indizieren und auffindbar machen.

Der Ansatz des Data Check-In besteht darin, Metadaten (welche die Daten und deren Zugriffsmöglichkeiten beschreiben) manuell für solche Objekte (Data-Source, z. B. bestimmte Anwendungen oder Portale, etc.) zu erfassen, für die eine automatische Detektierung der zugehörigen Metadaten als zu aufwendig erscheint oder nicht möglich ist, z. B. weil die Metadaten

nicht zugreifbar sind oder gar nicht existieren und eine automatische Ableitung solcher aus den Daten mit den existierenden Adaptoren als zu auf wendig erscheint.

Weitere Details

Hier geht es also primär um Datenquellen, die nicht geharvestet werden, sondern lediglich einmal oder selten manuell von einem Editor (oder per Import lokal bereits doch vorliegender Metadaten erfasst werden.

Schnittstelle(n)

Stellt bereit:

- Formulare zur möglichst einfachen und sicheren Erfassung von Metadaten
- API zur Erfassung von Metadaten

Benötigt:

- Create/Read/Update/Delete (CRUD-)Schnittstelle des Metadaten-Index

Qualitäts-/Leistungsmerkmale

Einfaches, verständliches, benutzerfreundliches und multi-userfähiges UI

2.1.4 Metadaten-Index

Beschreibung

Der Metadaten-Index enthält alle Metadaten, die über das Metadata Harvesting gefunden oder per Data Check-In eingepflegt werden (in einer für die Suche optimierten Form). Die Metadaten verweisen dabei auf die Originaldaten(s.5.1.1.4).

Weitere Details

Der Metadaten-Index muss einen effizienten Indizierungs- und Suchmechanismus bereitstellen. Er muss in der Lage sein, das verwendete Metadatenmodell zu interpretieren, bestimmte Suchparameter herauszuziehen und diese zu indizieren. Der Index muss zudem eine Volltext-Indizierung durchführen können.

Schnittstelle(n)

Bereitgestellt:

- MD-CRUD: Dieses ist die Create/Read/Update/Delete Schnittstelle, über die die Metadaten im Metadaten-Index von den verschiedenen Clients erzeugt, gesucht/gelesen, aktualisiert und gelöscht werden können. Diese Schnittstelle muss Operationen für die Suche nach Metadaten anbieten.
- MD-Search: Dieses ist die reine Such Schnittstelle (Read), über die die Metadaten im Metadaten-Index von den verschiedenen Clients gesucht/gelesen werden können.

Qualitäts-/Leistungsmerkmale

- Besonders hohe Verfügbarkeit und Skalierbarkeit da es sich um eine der zentralen Komponenten des Systems handelt
- Leistungsfähige Suchmaschine

2.1.5 Application

Beschreibung

Über die Application können die Nutzer*innen die Daten im umwelt.info Portal visualisieren und daraus neue Erkenntnisse ableiten. Dabei werden den Nutzer*innen unterschiedliche Anwendungen zur Visualisierung der Daten bereitgestellt.

Weitere Details

Für die Verwendung der Application müssen sich die Nutzer*innen am umwelt.info Portal registrieren. Nach der Registrierung stehen den Nutzer*innen die unterschiedlichen Anwendungen zur Verfügung.

Schnittstelle(n)

Über den Metadaten-Index können die Metadaten von den verschiedenen Anwendungen gesucht und gelesen werden. Diese Schnittstelle, ist die Verbindung zwischen der Application und den Metadaten-Index. Für Nutzer*innen wird dadurch ermöglicht, über die Metadaten den eigentlichen Datenbestand in eine der verschiedenen Anwendungen der Application visuell zu betrachten und mit anderen Daten zu kombinieren.

Bereitgestellt:

- Verschiedene Anwendungen

Benötigt:

- Metadaten-Index
- Identity Management

Qualitäts-/Leistungsmerkmale

Web-Applikation, Funktionalitäten bestehen aus einfachen Werkzeugen

Offene Punkte/Probleme/Risiken

Mit den unterschiedlichen Anwendungen in der Application soll es den Nutzer*innen möglich sein über die Metadaten die originalen Daten zu visualisieren, verarbeiten, sowie neue Daten abzuleiten. Die dazugehörigen Anforderungen sind noch nicht genau definiert und müssen vor einer Umsetzung im umwelt.info Portal näher spezifiziert werden.

Nach abgeschlossener Anforderungsanalyse des Projekts „Data Cube“, kann eine Anbindung von Komponenten des Data Cube in das umwelt.info System geprüft werden. Hierfür könnte eine Integration als Application stattfinden.

2.1.6 API

Beschreibung

Die Application Programming Interface (API) stellt Schnittstellen für die maschinelle Nutzung der Funktionen des Portals bereit. Beinhalten die Interoperabilitätsinterfaces, eine Identifizierungs- und Permalink-Schnittstelle für Datensätze, Suchschnittstellen und weitere, freigegebene Back-End-Dienste, die von der Benutzeroberfläche oder Nutzer*innen benötigt werden. Zum Beispiel benötigen Nutzer*innen zur automatischen Datenbereitstellung die HTTP-Schnittstelle des Data-Check-In, um eigene, datenbereitstellende Anwendungen anzubinden. Die über die

Benutzeroberfläche zur Verfügung gestellten Funktionen des Portals kommunizieren ebenfalls über die API den Back-End-Komponenten.

Weitere Details

Zur Nutzung einiger Funktionen, der APs, muss die API die Nutzer*innen mittels des Identity Managements authentifizieren und autorisieren. Zum Beispiel müssen Nutzer*innen für die Datenbereitstellung über die API der Data-Check-In Komponente genauso autorisiert sein, wie für die Nutzung über die UI des Data-Check-In. Die Schnittstellen und deren Zugriffssteuerung (wer darf wie zugreifen…) sind konfigurierbar.

Schnittstelle(n)

Bereitgestellt:

- OGC CSW AP ISO [4] (inkl. INSPIRE [5] (optional))
- OGC API Records
- PID-Service
- Hinzu kommen weitere APIs die von internen Komponenten bereitgestellt und nach außen (evtl. über eine Fassade) veröffentlicht werden sollen (z. B. der Data-Check-In)

Benötigt:

- Die APIs, die von internen Komponenten bereitgestellt werden und nach außen veröffentlicht werden sollen.
- Identity Management

Qualitäts-/Leistungsmerkmale

- Die bereitgestellten Schnittstellen folgen den jeweiligen Standards
- Die Dokumentation der weiteren Schnittstellen folgt dem OpenAPI 3-Standard für REST-Schnittstellen [6]
- Die API ist leicht erweiterbar, insbesondere für die Anbindung von weiteren internen Komponenten, deren <Schnittstell nach außen bereitgestellt werden sollen.

2.1.7 Identity Management

Beschreibung

Das Identity Management stellt alle notwendigen Funktionen bereit, damit Nutzer*innen den passenden Zugriff auf die Komponenten des umwelt.info Portals erhalten. Dazu zählen Identifikation und Authentifizierung von Nutzer*innen, Autorisierung des Zugriffs auf geschützte Ressourcen, und Verwaltung der Nutzer*innen durch die Portaladministrator*innen.

Weitere Details

Für die Authentifizierung verwenden die Nutzer*innen die Authentifizierung (personenbezogene Login-Informationen).

Es wird davon ausgegangen, dass für die Realisierung des Identity Management bestehende Frameworks und Produkte eingesetzt werden. Für diese bestehenden Produkte sollte es regelmäßige Aktualisierungen geben, wenn Schwachstellen in der Software gefunden werden.

Für die Authentifizierung und das Single Sign-On müssen verbreitete Standardprotokolle von der Software zur Verfügung stehen: OpenID Connect, SAML2, Central Authentication Service (CAS) Protokoll.

Schnittstelle(n)

Bereitgestellt:

- Standardprotokolle für Authentifizierung / Single Sign-On
- UI für die Anmeldeseite
- UI für die Profilverwaltung für angemeldete Nutzer*innen
- Verwaltung der Nutzer*innen durch Administrator*innen (Nutzer löschen, Rollen zuweisen)
- Definition von Zugriffsregeln durch Administrator*innen
- Bereitstellung von Nutzerinformationen auf Basis eines Security Tokens für Komponenten des umwelt.info Portals

Qualitäts-/Leistungsmerkmale

Unterstützung von Standard-SSO Protokollen

Offene Punkte/Probleme/Risiken

Das Identity Management muss einfach mit den anderen Komponenten des umwelt.info Portals integrierbar sein. Da Komponenten wie das CMS evtl. bereits auf anderen Produkten basiert, schränkt dies die Optionen für die Realisierung des Identity Managements wahrscheinlich ein.

Bezüglich der UI für die Profilverwaltung und der Nutzerverwaltung (Verwendung durch Administratoren) ist noch offen, welche Schnittstellen durch die Identity-Management-Komponente und welche durch das CMS realisiert werden.

2.1.8 Metadata Quality Assessment (MQA)

Beschreibung

Für die Validierung der Metadaten und zur Erhebung von Statistiken zur Qualität der im Metadaten-Index gespeicherten Metadatensätze wird eine Metadata Quality Assessment (MQA)-Komponente benötigt. Der Validator sollte auch von den Datenbereitsteller*innen genutzt werden können, um Fehler in den eigenen Metadaten zu erkennen.

Als Vorbild für die MQA-Komponente des umwelt.info Portals dient die Metadata Quality Assessment Methodologie [7] des European Data Portals (EDP). Betrachtet werden dabei Qualitätseigenschaften die (wie im EDP) aus den „FAIR Guiding Principles for scientific data management and stewardship“ [8] abgeleitet und gemäß „Data Quality Vocabulary“ [9] gespeichert werden. Die MQA-Komponente ermittelt die Qualitätseigenschaften: Auffindbarkeit, Zugänglichkeit, Interoperabilität, Wiederverwendbarkeit und Kontext. Diese Eigenschaften sind ebenfalls angelehnt an die Methodik des EDP.

Weitere Details

Die meisten Qualitätseigenschaften beziehen sich auf Felder der Metadatensätze zum Zeitpunkt der Speicherung, deren Validität sich über die Zeit nicht ändert. Felder, die den Zugang zu Daten beschreiben, müssen dagegen regelmäßig geprüft werden, da zum Beispiel verlinkte Distributoren ihren Dienst zwischenzeitlich eingestellt haben könnten.

Bei der Prüfung wird insbesondere betrachtet, welche Felder befüllt sind (z. B. Schlüsselwörter) und ob Angaben korrekt gemacht wurden (z. B. HTTP-Statuscode bei Zugriff über URL ist gleich 200):

- Auffindbarkeit (z. B. Schlüsselwörter, Kategorien)
- Zugänglichkeit (z. B. Erreichbarkeit der Access-URL)
- Interoperabilität (z. B. mediaType)
- Wiederverwendbarkeit (z. B. Lizenzangaben, Kontaktinformationen)
- Kontext (z. B. Rechte, Dateigröße, Änderungsdatum)

Für diese Qualitätseigenschaften werden jeweils bestimmte Felder geprüft. Für jede Qualitätseigenschaft ist zudem jeweils eine Metrik definiert, die sich aus der Gewichtung der Prüfergebnisse für die einzelnen Felder berechnet. Die Prüfergebnisse tragen dann jeweils mit ihrer Gewichtung zur Bewertung bei.

Die Bewertung der im Metadaten-Index gespeicherten Metadaten wird fortlaufend berechnet und lässt sich über die MQA-Komponente als Überblick über die Qualitätseigenschaften abrufen. Im Einzelnen wird bereitgestellt:

- Ein Gesamtüberblick über alle Daten im Metadaten-Index
- Ein Überblick pro Quelle
- Eine Einzelbewertung
- Eine Detailauswertung für jede Funktion und Qualitätseigenschaft

Schnittstelle(n)

Benötigt:

Metadaten-Index

Qualitäts-/Leistungsmerkmale

Aktualität und Umfang der bereitgestellten Auswertungen

Offene Punkte/Probleme/Risiken

Einschränkung: Die Qualität der Metadaten beschreibt nicht die Qualität der Daten selbst. Diese kann mit der MQA-Komponente nicht direkt ermittelt werden. Indirekte Indikatoren können lediglich auf besonders „interessante“ oder „wertvolle“ Daten hindeuten. Diese könnten zum Beispiel aus besonders häufig geklickten Downloadlinks oder häufig in Favoriten gemerkte Metadatensätze ermittelt werden.

2.1.9 Nutzertracking

Beschreibung

Für die Erhebung von Statistiken zur Nutzerbewegung im umwelt.info Portal, wird eine Nutzertracking-Komponente eingesetzt. Dadurch sollen wesentliche Informationen gesammelt werden, welche helfen das umwelt.info System für die Nutzenden stetig zu verbessern und gezielte Anpassungen zu entwickeln. Es gibt zahlreiche Software zur Erhebung von Nutzerstatistiken; die weitverbreitetsten sind Google Analytics und Matomo. Solche Softwareprodukte verfügen über ein umfangreiches Set an Nutzertracking Tools und bieten sich für eine Verwendung im umwelt.info Portal an, stellen aber keine Verbindlichkeit dar.

Weitere Details

Es gibt unterschiedliche Arten von Nutzerstatistiken, die mittels der zur Verfügung stehenden Tools in der Software erfasst werden können. Nachfolgend ist eine Auswahl von Funktionen aufgeführt, die für das Nutzertracking im umwelt.info Portal von Bedeutung sein könnten.

- Grundlegende Metriken – ermöglicht die Erfassung von Metriken zu Besuchern, Berichte, Standort, Gerät, Betriebssystem und Verhaltensweisen [10].
- Website-Aktionen – ermöglicht die Verfolgung des Verhaltens auf den einzelnen Webseiten
- Ziel – ermöglicht die wichtigsten gesetzten Ziele auf den Webseiten und misst deren Erfolg
- Trichter – ermöglicht Trichterprozesse, um Abbrüche zu messen und die Konversionsraten zu erhöhen
- Multi-Attribution – ermöglicht die Messungen der ersten und letzten digitalen Kanäle
- A/B-Tests – ermöglicht Testdurchführungen, um herauszufinden, welche Inhalte den Zielgruppen entsprechen
- Benutzerdefinierte Berichtserstattung – ermöglicht die Kombination von Metriken aus unterschiedlichen Funktionen, um ein besseres Verständnis der Nutzer*innen zu bekommen

Schnittstelle(n)

Bereitgestellt:

- Dashboards
- API

Benötigt:

- Integration mit CMS (Nutzung von UI-Komponenten)
- Integration mit API (Nutzung des Backend)
- Zugriff auf Logdateien (Sonstige Statistiken)

Qualitäts-/Leistungsmerkmale

Erfassung und Auswertung aktueller Nutzerstatistiken, welche in einer dafür eigens vorhandenen Nutzeroberfläche zusammengetragen werden.

Offene Punkte/Probleme/Risiken

Bei der Erfassung von Nutzerstatistiken sollte berücksichtigt werden, dass es dabei zur Erfassung von personenbezogenen Daten kommt und diese darüber informiert werden müssen. Es sollte dabei stets eine Prüfung der Software vorgenommen werden und ob diese sich an die Richtlinien der Datenschutz-Grundverordnung (DSGVO) hält.

2.1.10 Data-Source

Beschreibung

Die Data-Source kennzeichnet hier die Quellen aller Daten, die durch die Harvester bzw. Crawler durchsucht werden und auf die die Metadaten im Metadaten-Index des umwelt.info Portals verweisen. Es kann sich aber auch um eine Datenquelle handeln, die manuell mittels Metadaten beschrieben wird.

Weitere Details

keine

Schnittstelle(n)

Bereitgestellt: Die hier potenziell vorkommenden Schnittstellen sind im Kapitel 3.2 im Umsetzungskonzept beschrieben.

Qualitäts-/Leistungsmerkmale

Dieses liegt nicht in der Hand des umwelt.info Portals.

Offene Punkte/Probleme/Risiken

Risiken:

Es kann sich bei den Data-Sources um Datenquellen mit unbekannten oder komplexen Interfaces und Datenformaten handeln, die zudem schlecht dokumentiert sein können.

Die Performance und Verlässlichkeit der Schnittstelle sowie die Qualität der Daten der Data-Source kann durchaus schlecht sein, was auf das Harvesting stark negative Einflüsse haben kann.

2.1.11 Recommender System (RS) – Basisimplementierung (Inhalts-/Indexbasiert)

Beschreibung

Um Metadatensätze als Empfehlungen für Nutzer*innen erhalten zu können wird ein Recommender System (RS) benötigt. Diese Komponente nimmt den aktuellen Nutzer*innen-Kontext als Eingabe entgegen und liefert als Ausgabe Empfehlungen, welche Metadatensätze zum gegebenen Kontext passen. So werden Empfehlungen etwa für einen einzelnen Metadatensatz oder für mehrere Metadatensätze als Kontext gegeben (etwa für den aktuell betrachteten oder für die ersten drei Metadatensätze einer Suchergebnisliste).

Weitere Details

In einer Basisumsetzung kann die Funktionalität des Index genutzt werden. Dabei wird für einen gegebenen Kontext eine Suche im Index durchgeführt. Dazu kann auch auf dafür vorgesehene Funktionen des Index zurückgegriffen werden. In Solr kann dazu etwa die „MoreLikeThis“ Komponente genutzt werden.

https://solr.apache.org/guide/8_11/morelikethis.html

Je nach Algorithmus können neben dem aktuell betrachteten Metadatensatz als Kontext auch weitere Datengrundlagen betrachtet und weitere Werkzeuge eingesetzt werden.

Optionen:

- Merklisten und Collaborative Filtering (siehe 2.2.1)
- Nutzung von Linked Data und Semantischen Analysen (siehe Kapitel 4 im Umsetzungskonzept)

Schnittstelle(n)

Bereitgestellt:

Recommender-API (gibt Empfehlungen für einen gegebenen Nutzer*innen-Kontext)

Benötigt:

Metadaten-Index

Qualitäts-/Leistungsmerkmale

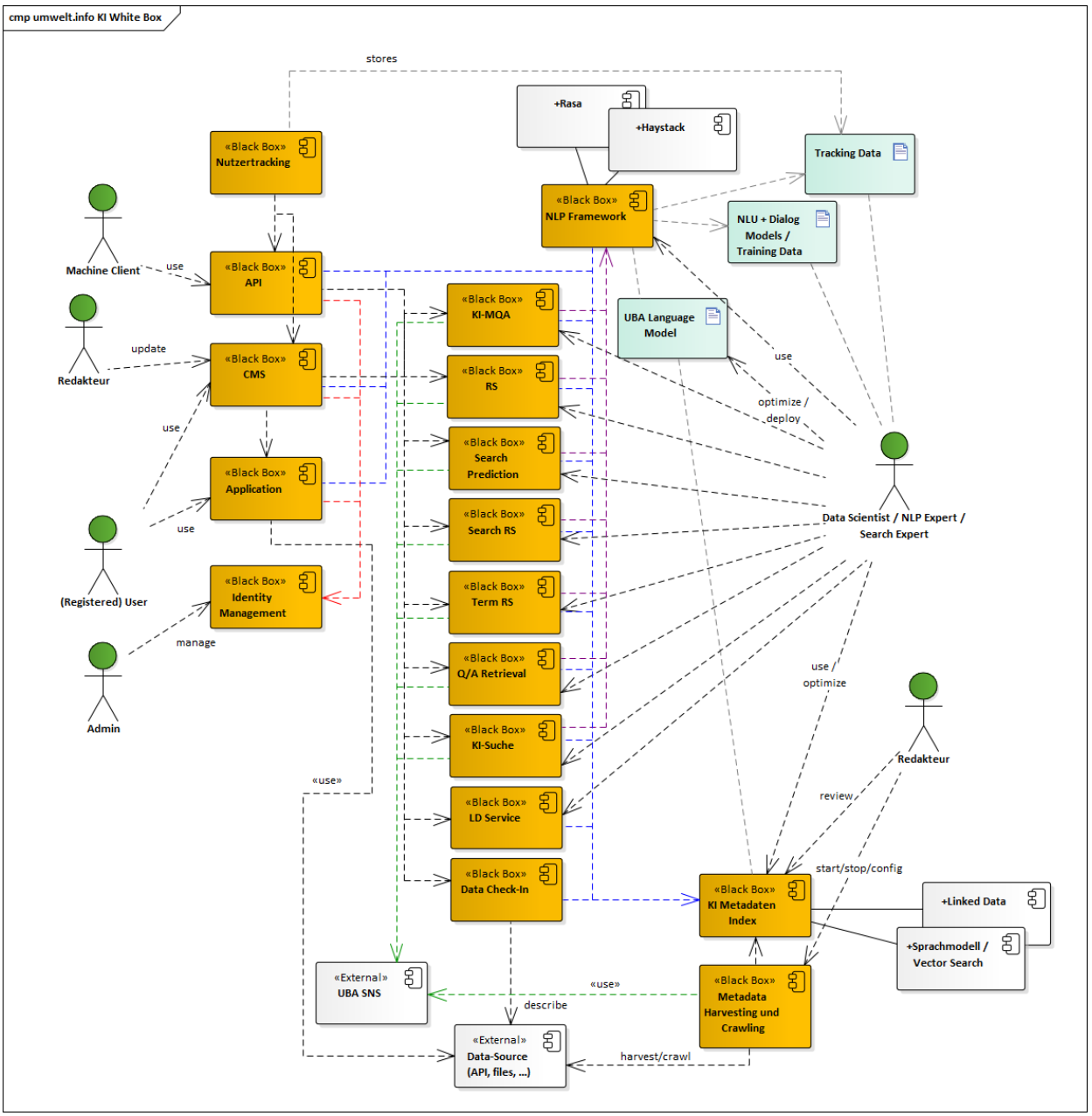
- Es werden Metadatensätze empfohlen, die für Nutzer*innen und den gegebenen Kontext relevant sind
- Performant
- Einfache Umsetzung

Offene Punkte/Probleme/Risiken

Die Optimierung dieser Komponente findet im Rahmen der Projektbegleitung statt. Hierbei sollten auch die möglichen Optionen für eine Weiterentwicklung und deren Erfolg laufend geprüft werden

2.2 Ebene 1 umwelt.info: KI und Linked Data Optionen (White Box)

Abbildung 3: Ebene 1 umwelt.info: KI und Linked Data Optionen (White Box)



Quelle: eigene Darstellung, con terra GmbH

2.2.1 Recommender System (RS) – Merklisten und Collaborative Filtering

Beschreibung

Neben inhaltsbasierten Vorschlägen können auch nutzungsbasierte Ansätze entwickelt werden. Ein Mehrwert wird dabei für die Nutzenden erzielt, wenn Empfehlungen relevante Metadatensätze enthalten, die für andere Nutzende mit ähnlichen Interessen ebenfalls relevant sind. Dieses

Potenzial ist für umwelt.info interessant, da „ähnliche Interessen“ ein Konzept ist, das sich nicht allein durch die Analyse der Inhalte im Index erschließen lässt.

Dabei gilt, dass sich nutzungs- und inhaltsbasierte Ansätze nicht gegenseitig ausschließen. Vielmehr sollte es das Ziel sein, beide Ansätze sinnvoll zu integrieren und für den vorliegenden Anwendungsfall eine Lösung fortlaufend zu optimieren.

Weitere Details

Ein einfacher Nutzer*innen-zentrierter Algorithmus kann ermitteln, welche Metadatensätze in den Favoriten der Nutzer*innen, zusammen mit dem aktuell betrachteten Metadatensatz gespeichert sind und wie oft. Aus den so gefundenen, potenziell relevanten Metadatensätzen werden dann Empfehlungen ermittelt und zurückgegeben.

Außer dieser einfachen Auswertung von Favoriten kann auch „Collaborative Filtering“ als Algorithmus eingesetzt werden. Dabei ist die grundlegende Idee die gleiche wie bei der Auswertung der Listen. Es wird über die Beziehung von Nutzer*innen zu Metadatensätzen ermittelt, welche anderen Nutzer*innen ähnliche Interessen aufweisen, um darüber weitere relevante Metadatensätze zu finden.

Dabei können auch andere Nutzungsdaten analysiert werden. In umwelt.info könnte so auch mittels Zählung von Klicks der Nutzer*innen die Beziehungen der Nutzer*innen zu Metadatensätzen ermittelt werden. Außerdem lässt sich das Verfahren auch auf andere Zieleigenschaften anwenden, etwa um Schlagworte vorzuschlagen.

Wenn ein Nutzungssignal (etwa getrackte „Klicks“, vgl. 2.1.9) vorliegt, lässt sich über die Suchmaschine das Ableiten von ähnlichen Metadatensätzen mit einer speziellen Anfrage als Collaborative Filtering realisieren. Ein Beispiel mit Solr als Suchmaschine wird in Kapitel 16, in [11] beschrieben.

Schnittstelle(n)

Siehe 2.1.11

Qualitäts-/Leistungsmerkmale

Verbesserung der Vorschläge durch Analyse von Nutzungsdaten.

Offene Punkte/Probleme/Risiken

Bei einem Abgleich des aktuellen Metadatensatzes mit den vorhandenen Favoriten ist die Qualität der Empfehlungen von den erstellten Favoriten abhängig. Initial werden nur wenige Favoriten von Nutzer*innen befüllt sein. Daher kann die Qualität der Empfehlungen, insbesondere am Anfang, nur schwer gemessen und verbessert werden.

Generell gilt, dass initial (zu Beginn der Einführung von umwelt.info, sowie für neue Nutzende) nur wenige Nutzungsdaten vorhanden sein können. Daher sind zu Beginn auch nur wenige (sinnvolle) Empfehlungen auf dieser Basis zu erwarten. Algorithmen, die diesen Ansatz verfolgen benötigen einen längeren „Beobachtungszeitraum“ und eine breite Befüllung des „Suchraums“, um sinnvolle Empfehlungen zu generieren.

Als Lösung für dieses Problem sollte zunächst ein Inhaltsbasierter Ansatz entwickelt werden (siehe 2.1.11). Erweiterte Algorithmen können in der Projektbegleitung getestet und kombiniert werden.

2.2.2 KI-Suche

Beschreibung

Eine vorgeschaltete Komponente für die Suchfunktion ist die KI-Suche. Dabei ist diese Komponente für die Verbesserung der Suche zuständig: Die Vorverarbeitung von Suchanfragen (etwa mittels NLP-Methoden), Suchanfragenoptimierung (etwa für die Gewichtung der Felder hinsichtlich der Relevanz der Suchtreffer) und für die Ermittlung weiterer Informationen, die ggf. für die Unterstützung des Suchenden (bzw. für die UI) hilfreich sind.

Die Hauptaufgabe der KI-Suche ist es, die Relevanz der einzelnen Treffer hinsichtlich einer Sucheingabe zu optimieren. Für das Ranking wird i. a. zunächst der Standardalgorithmus BM25, des Suchindexes genutzt. Weitere Algorithmen bieten sich an und sollten im Rahmen der Tätigkeit des „Relevance Engineering“, laufend optimiert werden. Bestimmte Optimierungen lassen sich dabei auch automatisieren bis hin zu Ansätzen der Automatisierung des Relevance Engineering selbst.

Weitere Details

Zunächst können mittels Vorverarbeitung von Texteingaben Verbesserungen für die Suche erzielt werden. Dabei werden Suchanfragen analysiert, um etwa Konzepte und Kontextinformationen in der Sucheingabe zu erkennen. So kann als Kontextinformation auch das Ziel einer Suche (etwa ob eine konkrete Frage gestellt wurde oder ob der Nutzer eine Liste von Datensätzen zu einem Thema als Ergebnis wünscht) mittels Intent-Erkennung bestimmt werden, um das erkannte Ziel bei der Anfrage an den Metadaten-Index zu berücksichtigen.

Eine KI-Suche kann für eine einfache Suchanfrage weitere Informationen ermitteln und die Suche damit anreichern oder verbessern. Dazu kann die KI-Suche auch NLP-Frameworks nutzen, um etwa Intent-Erkennung zur Bestimmung der Art der Suche.

Darüber hinaus lässt sich auch das Dialogsystem eines Chatbot-Frameworks für die Steuerung von Assistenzfunktionen nutzen, um diese im richtigen Moment anzuzeigen.

Bei der Entwicklung dieser Komponente können Verarbeitungsmethoden und Anfragetechniken wie unter Kapitel 9.6 (Künstliche Intelligenz) im Umsetzungskonzept beschrieben, genutzt werden, um eine Optimierung des Rankings zu erreichen. Darüber hinaus existieren „Machine Learning“-Verfahren zur Automatisierung der Optimierung des Rankings. Der von der Suchmaschine Solr bereitgestellte Ansatz hierzu heißt zum Beispiel „Learning to Rank“.

https://solr.apache.org/guide/8_11/learning-to-rank.html

Bei einem solchen LTR-Verfahren ist jedoch zu beachten, dass Nutzungsinformationen als Trainingsdaten eingesetzt werden müssen (vgl. 9.6.4). Außerdem müssen diverse Seiteneffekte (etwa „Bias“) berücksichtigt werden. Die Kapitel „Building learning to rank training data from user clicks“ und „Overcoming bias in learned relevance models“ in [12] beschreiben diese Methode und den Umgang mit realen Daten.

Mögliche Optionen bei der Entwicklung dieser Komponente sind:

- Basisumsetzung (Suchanfrage weiterreichen, nur Informationen zur Suche ermitteln, die für die UI benötigt werden)
- Verbesserung der Suche mittels Linked Data- und NLP-Methoden
- Einsatz eines Sprachmodells
- Entwicklung eines LTR-Ansatzes

Schnittstelle(n)

Bereitgestellt:

- Eingabe: Suchparameter
- Ausgabe: Suchergebnisliste

Benötigt:

- Metadaten-Index

Qualitäts-/Leistungsmerkmale

Optimiertes Ranking

2.2.3 Linked Data (LD) Service

Beschreibung

Eine wichtige Funktion, die der LD-Service bereitstellt, ist das Selektieren eines Sub-Graphen aus dem Knowledge Graphen auf der Basis einer Anfrage. Mit diesem Sub-Graphen lässt sich dann ein Mashup erzeugen. Dieses kombiniert verschiedene Informationen in einer Web-Seite, um einen schnellen Überblick über ein Thema zu bieten, anstelle es ausschließlich dem Suchenden zu überlassen, selber die verlinkten Informationen / Dokumente sukzessive aufzusuchen, um sich daraus die wichtigsten Informationen extrahieren zu müssen.

Weitere Details

Eine mögliche Lösung für das Selektieren eines Sub-Graphen könnte etwa darin bestehen, hierfür vom System nach solchen DCAT-AP Ressourcen (DCAT-AP Graphen) im Metadaten-Index zu suchen, die möglichst gut auf die Suchanfrage passen (z. B. hohe Ähnlichkeit bei Begrifflichkeiten, Raumbezug, Zeit, …). Beim Ranking in der Suchmaschine sollte dazu berücksichtigt werden, dass Ressourcen, die viele eingehende Links anderer Ressourcen aufweisen und zudem selbst möglichst viele Links (etwa Verweise auf Bilder etc.) besitzen, in der Ergebnisliste möglichst weit nach oben wandern. Dann muss der Sub-Graph, der durch die Attribute des am höchsten ge-rankten Treffers sowie der mit diesem Treffer direkt verlinkten Ressourcen aufgespannt wird, kopiert und an den Anfrager zurückgeliefert werden.

Schnittstelle(n)

Bereitgestellt:

- Einige spezielle LD-Operationen, wie etwa „Selektieren eines Subnetzes aus dem Knowledge Graphen“

Benötigt:

- Search-API: Native Schnittstelle der Suchmaschine zur Suche nach indizierten Dokumenten.
- SPARQL

Qualitäts-/Leistungsmerkmale

Ranking muss für das Finden des besten Graphen optimiert und feinjustiert werden.

Offene Punkte/Probleme/Risiken

Es ist offen, wie entschieden wird, wie viele Links (und von welchem Typ diese sein müssen) verfolgt werden, um den Sub-Graphen aufzubauen.

2.2.4 Search Prediction

Beschreibung

Diese Komponente bietet eine spezielle Schnittstelle zum Metadaten-Index. Ihre Aufgabe ist die Bereitstellung einer Suchvorhersage für eine unvollständige Sucheingabe. Vorhersagen werden auf Grundlage der gespeicherten Metadaten im Index oder weiteren, aufbereiteten Datengrundlagen gegeben.

Weitere Details

Eine einfache Basisumsetzung kann mit einer entsprechenden Funktion des Index realisiert werden, etwa mit Solr's „Suggester"-Komponente, die in verschiedenen Ausprägungen zur Verfügung steht und dabei z. B. zusätzliche Felder oder sog. „n-grams" nutzt. Dabei können vorhandene Textfelder des Metadaten-Index abgefragt werden oder eine eigene Sammlung von Suchvorschlägen in einem dedizierten Suchvorschlagindex gepflegt werden (siehe https://solr.apache.org/guide/8_11/suggester.html)

Weitere Optionen:

- Es ist möglich, die Dienste des SNS (etwa den Umwelt Thesaurus oder die SNS-Chronik) für die Befüllung eines Suchvorschlagindexes zu nutzen, um Wartungsaufwände zu minimieren.
- Während eine einfache Suchvorhersage mittels Suggester-Funktionalität des Index realisiert werden kann, bietet die Analyse von Eingaben und der Textfelder des Index (etwa der Überschrift und Kurzbeschreibung) mittels NLP- und Linked Data-Methoden signifikantes Verbesserungspotenzial für die Suchvorhersage. Dabei können die unter Kapitel 4 (Künstliche Intelligenz) im Umsetzungskonzept beschriebenen Konzepte eingesetzt werden.

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle zur Abfrage von Vorhersagen
- Eingabe: Unvollständige Sucheingabe für den Suchschlitz und ggf. weitere Kontextinformationen
- Ausgabe: Vollständige Sucheingabe für den Suchschlitz

Benötigt:

- Metadaten-Index

Qualitäts-/Leistungsmerkmale

- Performant und skalierbar im Betrieb (die Anfragehäufigkeit beträgt ein Vielfaches von der des Metadaten-Index)
- Betriebskosten sind abhängig von den gewählten Optionen.

- Der Einsatz dieser Funktion ist mit geringem Wartungsaufwand verbunden. Die Qualität der Ausgaben sollte laufend geprüft werden, um ggf. nachzubessern.

Offene Punkte/Probleme/Risiken

Es ist nicht automatisch sichergestellt, dass eine gepflegte Sammlung von Suchvorschlägen immer zu Suchergebnissen führt, da diese technisch unabhängig vom eigentlichen Metadaten-Index ist. Die Prüfung der Treffer zu Listeneinträgen sollte daher Bestandteil der Pflege von Suchvorschlagslisten sein.

2.2.5 Search RS

Beschreibung

Die Aufgabe des Search RS (Suchvorschlagssystem) ist die Bereitstellung von Suchvorschlägen für eine abgeschlossene Sucheingabe. Das Ziel der Search RS Komponente ist es, „ähnliche Suchen" vorzuschlagen. Vorschläge werden wie bei der Search Prediction auf Grundlage der gespeicherten Metadaten im Index oder weiteren, aufbereiteten Datengrundlagen gegeben.

Weitere Details

Strukturell ähnelt die Search RS Komponente der Search Prediction Komponente. Unterschiedlich sind die Eingaben und Ausgaben der beiden Komponenten, sowie die Anforderung an die Performanz, die für die Search RS Komponente, im Vergleich zur Suche, nicht erhöht ist. Es bestehen die gleichen Optionen für die Entwicklung wie für die Search RS Komponente (siehe 2.2.4).

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle zur Abfrage von Vorschlägen
- Eingabe: Vollständige Sucheingabe, ggf. weitere Filter und weitere Kontextinformationen
- Ausgabe: Links zu ähnlichen Suchen

Benötigt:

- Metadaten-Index

Qualitäts-/Leistungsmerkmale

- Performant und skalierbar im Betrieb (entsprechend der Suche auf dem Metadaten-Index).
- Betriebskosten sind abhängig von den gewählten Optionen.
- Der Einsatz dieser Funktion ist mit geringem Wartungsaufwand verbunden. Die Qualität der Ausgaben sollte laufend geprüft werden, um ggf. nachzubessern.

Offene Punkte/Probleme/Risiken

Es ist nicht automatisch sichergestellt, dass eine gepflegte Sammlung von Suchvorschlägen immer zu Suchergebnissen führt, da diese technisch unabhängig vom eigentlichen Metadaten-Index ist. Die Prüfung der Treffer zu Listeneinträgen sollte daher Bestandteil der Pflege von Suchvorschlagslisten sein.

2.2.6 Term RS

Beschreibung

Die Aufgabe der Terms RS Komponente ist die Bereitstellung von Begriffen, die zu einer gegebenen Suche zu einer Einengung oder Erweiterung des Suchraums führen.

Weitere Details

Strukturell ähnelt die Term RS Komponente der Search RS Komponente. Der grundlegende Unterschied ist dabei der Bezug zu den Indizierten Metadaten. Für die Term RS-Komponente müssen die vorgeschlagenen Begriffe aus den Indexierten Textfeldern abgeleitet und zu einer „Erweiterung“ bzw. „Verfeinerung“ der Treffermenge führen. Eine mögliche Umsetzung basiert auf einer semantischen Analyse der Textfelder und wird im Kapitel 9.6 (Künstliche Intelligenz) im Umsetzungskonzept, als querschnittliches Konzept beschrieben. Darüber hinaus bestehen die gleichen Optionen wie für die anderen suchnahen Komponenten (siehe 2.2.4).

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle zur Abfrage von Vorschlägen
- Eingabe: Vollständige Sucheingabe, ggf. weitere Filter und weitere Kontextinformationen
- Ausgabe: Liste mit Begriffen zur Anpassung der Suche

Benötigt:

- Metadaten-Index

Qualitäts-/Leistungsmerkmale

- Performant und skalierbar im Betrieb (entsprechend der Suche auf dem Metadaten-Index).
- Betriebskosten sind abhängig von den gewählten Optionen.
- Der Einsatz dieser Funktion ist mit geringem Wartungsaufwand verbunden. Die Qualität der Ausgaben sollte laufend geprüft werden, um ggf. nachzubessern.

Offene Punkte/Probleme/Risiken

Die hier vorgeschlagene Verwendung von semantischen Abfragen auf dem Index zur Begriffsermittlung ist ein erster, konzeptioneller Lösungsansatz. Die Effektivität dieses Ansatzes hinsichtlich des Verwendungszwecks „Verfeinern“ und „Erweitern“ ist noch offen.

2.2.7 Q/A Retrieval

Beschreibung

Die Q/A Retrieval-Komponente findet für eine (als Frage formulierte) Sucheingabe in den indexierten Textfeldern der Metadaten eine Textpassage, die als konkrete Antwort auf eine Frage gegeben werden kann.

Weitere Details

Antworten werden auf Basis von indexierten Texten ermittelt. Textfelder im Suchindex von umwelt.info sind insbesondere Überschriften und Beschreibungen (vgl. 5.1.1.4). Es wäre aber auch

denkbar für diese Komponente weitere Textdaten zu indexieren oder einen eigenen Index vorzuhalten.

Die Schritte für das Auffinden einer Antwort sind:

- Suche im Index nach relevanten Dokumenten
- Re-Ranking
- Passage Retrieval

Dieses Verfahren basiert hauptsächlich auf Fähigkeiten des Indexes, wie etwa re-ranking Klauseln in einer Suchanfrage oder „Highlighten“ für das Passage Retrieval (https://solr.apache.org/guide/8_11/highlighting.html https://solr.apache.org/guide/8_11/query-re-ranking.html).

Moderne Q/A-Systeme nutzen zudem auch Sprachmodelle zur Vektorisierung von Textfragmenten (siehe. 5.6.3 und 2.11). Im NLP Framework Haystack etwa besteht eine entsprechende Pipeline aus den Komponenten „Retriever“ und „Reader“ (https://haystack.deepset.ai/components/ready-made-pipelines#extractiveqapipeline).

Während dieses Prozesses können auch verschiedene weitere Techniken zum Einsatz kommen (vgl. Kapitel 9.6 (Künstliche Intelligenz) im Umsetzungskonzept). Zum Beispiel kann eine Intent-Erkennung zur Klassifikation von Eingaben in verschiedene Fragetypen genutzt werden, um verschiedene Antworttypen besser berücksichtigen zu können.

Schnittstelle(n)

Bereitgestellt:

- Q/A Schnittstelle
- Eingabe: Suchanfrage
- Ausgabe: Antwortpassage

Benötigt:

- Metadaten-Index
- NLP-Framework

2.2.8 KI-MQA

Beschreibung

Die KI-MQA-Komponente ist eine Erweiterung der MQA-Komponente um Möglichkeiten zur Nutzung von NLP-Techniken sowie der Möglichkeiten, weitere Anfragen gegen den Index zu stellen oder die Daten selbst zu laden.

Weitere Details

Die KI-MQA kann

- die Daten selbst nutzen, um Metadaten automatisch zu erzeugen und um die erkannten Metadaten mit den angegebenen Metadaten abzugleichen.

- die Textfelder in einem Metadatensatz auswerten, um auf Grundlage der anderen Felder ein neues Feld zu befüllen.

Zudem können der SNS und weitere Thesauri eingesetzt werden, um erkannte Begriffe wieder auf Konzepte im Index zu mappen (etwa die Erkennung von Schlagworten im Umweltthesaurus und die Erkennung der geografischen Region mittels des Geothesaurus).

Auf dieser Basis lässt sich (perspektivisch) ein „Suggester“ für die Metadaten-Eingabemaske entwickeln.

Schnittstelle(n)

Benötigt für Qualitätssicherung:

- Zugriff auf die Daten
- (KI-)Metadatenindex
- NLP-Framework
- SNS

Benötigt für Suggester:

- Zu vervollständigenden Metadatensatz

Qualitäts-/Leistungsmerkmale

- Durch den Zugriff auf die Quellendaten ergeben sich Möglichkeiten für die Qualitätskontrolle. So wird auch eine „Anomalie Erkennung“ in (strukturierten) Quellen-Daten möglich.
- Es wird eine Konsistenzüberprüfung der Felder ermöglicht. So lässt sich zum Beispiel prüfen, ob der Titel zur Kurzbeschreibung passt. Es kann so auch versucht werden, in den Freitextfeldern Werte für andere strukturierte Felder zu finden und diese zu mappen.
- Es können mehrere Metadatensätze, die den gleichen Datensatz (unterschiedlich) beschreiben, gefunden werden.
- Es könnte versucht werden, die Metadatensätze zu bewerten, etwa mit einer Art Kennwert oder Entropie. Hierbei sollen Aspekte berücksichtigt werden wie Wortaussagekraft (etwa im Index), Satzlänge, Verständlichkeit, Füllwörter, etc.

Offene Punkte/Probleme/Risiken

Es ist unklar, inwieweit sich etwa eine Anomalie Erkennung in den Quellendaten generalisieren (also auf alle möglichen Quellendaten anwenden) lässt. Jeder Datentyp (oder sogar jede Datenquelle) wirft möglicherweise eigene technische Fragen auf, die individuell gelöst werden müssen.

Der Zugriff auf die Quellendaten erfordert den Einsatz von entsprechenden Ressourcen, oder kann nur in geringen Umfang erfolgen (zum Beispiel nicht für das Harvesten, aber für die Metadateneingabemaske).

2.2.9 NLP Framework

Beschreibung

Für viele Komponenten werden in umwelt.info Methoden aus dem Bereich des Natural Language Processing (NLP) benötigt. Beispiele sind etwa die Verarbeitung von Suchanfragen, Bereitstellung von Vorschlagsfunktionen oder die Analyse von Textfeldern zur Qualitätssicherung). Dafür sollte ein geeignetes NLP-Framework genutzt werden, welches es Entwicklern und Data Scientists erleichtert, Funktionen und Analysen für umwelt.info zu entwickeln und diese im gegebenen technologischen Rahmen einzusetzen.

Produkte die in umwelt.info eingesetzt werden könnten sind zum Beispiel Rasa, um etwa das Auswerten von (zusammenhängenden) Trackingdaten zu erleichtern, oder Haystack, um bestimmte Indexfunktionen (wie „dense vector search“) zu nutzen.

Weitere Details

Bei der Realisierung verschiedener Komponenten werden diverse Funktionen aneinandergekettet (etwa die Auswertung einer Anfrage mittels klassischer NLP-Methoden und der eigentlichen Abfrage gegen den Metadaten-Index). Solche „Pipelines“ lassen durch den Einsatz eines passenden Frameworks leichter entwickeln, abstrahieren und wiederverwenden. Im Betrieb zeichnet sich ein geeignetes Framework außerdem dadurch aus, dass die eingesetzte Technologie unterstützt wird (Suchmaschinen) und entwickelte Pipelines und trainierte Modelle wartbar sind.

Schnittstelle(n)

N/A

Qualitäts-/Leistungsmerkmale

- Erleichtert die Entwicklung und den Einsatz von Pipelines
- Es können klassische NLP-Methoden und Sprachmodelle genutzt werden

2.2.10 Haystack

Beschreibung

Haystack ist ein „Such-Framework“, das in umwelt.info die Rolle des NLP-Frameworks einnehmen kann. Es kann insbesondere für den Einsatz von Q/A-Retrieval- bzw. Sprachmodellbasierter Suche eingesetzt werden.

Weitere Details

Der Fokus von Haystack liegt laut Herstellerangabe auf der Entwicklung von Q/A-Systemen. Eine Pipeline für den Q/A-Einsatz wird unter 2.2.7 beschrieben.

Haystack unterstützt dabei insbesondere auch die „semantisch augmentierte“ Suche unter Verwendung eines Sprachmodells. Dabei werden Vektorrepräsentationen von Textfragmenten genutzt, die bei der Indexierung mit den Datensätzen gespeichert werden und im Betrieb bei der Suche abgefragt werden (vgl. 2.11). Dabei unterstützt Haystack auch das „Nachtrainieren“ (auch „Fine-Tuning“) eines großen, vortrainierten Modells, wie etwa German BERT (https://www.deepset.ai/german-bert, https://haystack.deepset.ai/tutorials/fine-tuning-a-model).

Für die Suche auf den Indexierten Texten mittels Vektorrepräsentationen stellt Haystack die benötigte Funktion bereit. Es ist absehbar, dass diese Funktion in Zukunft zunehmend durch Indexkomponenten selbst übernommen wird (etwa mittels „Approximate Nearest Neighbour“-Suche in Solr oder Elasticsearch).

Für die Klassifikation von Sucheingaben stellt Haystack einen Klassifikator bereit, etwa zur Unterscheidung verschiedener Fragetypen (https://haystack.deepset.ai/tutorials/query-classifier).

Haystack unterstützt auch die Nutzung von Knowledge Graphen für die Aufgabe des „Question Answering“. Diese Funktion wird bisher nur mit Einschränkungen bereitgestellt. Für das Training eines eigenen Modells zur Übersetzung von natürlichsprachlichen Suchanfragen in SPARQL-Anfragen gibt es bisher keine Unterstützung (Stand: 13.06.2022) (https://haystack.deepset.ai/guides/knowledge-graph).

Qualitäts-/Leistungsmerkmale

- Intent Klassifikation für Fragetypen
- State of the Art + „out of the box“-Konfiguration
- Ermöglicht den Einsatz von Sprachmodellen
- Ermöglicht das nachtrainieren eines Sprachmodells
- Flexibel konfigurier- und anpassbare Pipelines
- Enthält ein „Annotation Tool“ zur Pflege von Q/A-Retrival Datasets
- Haystack ist Open Source

Offene Punkte/Probleme/Risiken

- Abhängig von der gewünschten Funktionalität kommt es zu Wartungsaufwänden für die Pflege eines natürlichsprachlichen Interfaces.
- Der Einsatz eines großen Sprachmodells (etwa ein speziell nachtrainiertes BERT-Modell) kann im Betrieb erhöhte Rechenkapazitäten benötigen.
- Bei der Nutzung von Knowledge Graphen ist das Problem der Übersetzung einer Textanfrage in eine SPARQL-Anfrage durch Haystack bisher ungelöst.

2.2.11 Rasa

Beschreibung

Rasa kann in umwelt.info als weiteres NLP-Framework eingesetzt werden, wobei es teilweise andere Aufgabenbereiche als Haystack abdeckt.

Rasa ist ein „Conversational AI-Framework“, das in umwelt.info eingesetzt werden kann, um NLP- und Dialogmanagement-Aufgaben zu übernehmen. Diese Aufgaben werden in der Rasa Default-Konfiguration mit Machine Learning-Methoden und deterministischen Regeln gelöst. Neben einer Konfigurationsdatei benötigt Rasa Trainingsdaten für die Dialogsteuerung und für das Natural Language Understanding.

Weitere Details

Dialoge sind in Rasa nicht auf Eingaben und Ausgaben in natürlicher Sprache (Text) beschränkt. Vielmehr handelt es sich bei einem Dialog in Rasa um einen Ablauf von „Events“, wobei es eingehende Events (Intents) und ausgehende Events (Actions) gibt.

Für die Implementierung eines Chatbots werden Intents und Actions als Dialoge aufgefasst. Nutzereingaben werden klassifiziert und Actions ausgeführt. Meist handelt es sich dabei um Textnachrichten an den Chatbot und Antworten von diesem.

Für umwelt.info können Intents für Sucheingaben erkannt werden und Actions können beliebige Funktionen sein. Somit kann statt einer textuellen Antwort, als Action zum Beispiel ein Button angezeigt werden, der in der UI eine Anpassung/Verbesserung der Suche bewirkt.

Rasa kann flexibel eingesetzt und angepasst werden. Insbesondere kann die Machine Learning-Pipeline vollständig konfiguriert werden, wobei auch ein eigenes Sprachmodell eingesetzt werden kann.

Neben dieser Flexibilität verfolgen die Entwickler von Rasa das Ziel „state of the art"-Techniken und Komponenten für die Entwicklung und den Betrieb von „Conversational AI-Assistenten" oder „Chatbots" bereit zu stellen.

Alle für den Betrieb benötigten Rasa-Komponenten werden vom Hersteller bereits für den Einsatz in Kubernetes vorbereitet.

Weitere Informationen: https://rasa.com/docs/rasa/, https://rasa.com/docs/action-server/events/, https://rasa.com/docs/rasa/arch-overview, https://rasa.com/docs/rasa/how-to-deploy

Schnittstelle(n)

Bereitgestellt:

- Eventbasierte Schnittstelle für Intents und Actions (Anbindung an API/UI)
- Nutzerschnittstelle für Pflege und Dialogentwicklung

Benötigt:

- Konfiguration
- Trainingsdaten
- Schnittstellen anderer Systemkomponenten für „Custom Actions"
- Optional: Spezielles Sprachmodell

Qualitäts-/Leistungsmerkmale

- Generelle Intent- und Entity Erkennung
- Werkzeuge für den gesamten „Conversational AI"-Lifecycle (Enterprise Lizenz)
- Open Source (Kernkomponente)
- Aktive Community
- Deployment in Kubernetes
- Training von eigenen Modellen für NLU- und Dialog mit eigenen Daten
- Einsatz von eigenen/nachtrainierten Sprachmodellen möglich

- Flexibel konfigurier- und anpassbare Pipelines
- Beispiele werden vom Hersteller bereitgestellt und gepflegt („out of the box“ und „state oft he art“)

Offene Punkte/Probleme/Risiken

Abhängig von der gewünschten Funktionalität kommt es zu Wartungsaufwänden für die Pflege eines natürlichsprachlichen Interfaces.

Der Einsatz eines großen Sprachmodells (etwa ein speziell nachtrainiertes BERT-Modell) kann im Betrieb stark erhöhte Rechenkapazitäten benötigen.

2.2.12 KI-Metadatenindex

Beschreibung

Der KI-Metadaten-Index ist eine Erweiterung des Metadaten-Index um Funktionen, die von den KI-Komponenten benötigt werden. Insbesondere sind dies die in Ebene 2 beschriebenen Optionen Linked Data (2.10) und Sprachmodell (2.11), sowie einzelne Funktionen, die zusätzlich von der Suchmaschine bereitgestellt werden (z. B. die in 2.2.4 beschriebene Liste von Suchvorschlägen).

Weitere Details

Siehe auch (Metadaten-Index Komponenten):

- 2.10, Ebene 2 umwelt.info – Metadata-Index – Option: Linked Data und Spatial Data on the Web (White Box)
- 2.11, Ebene 2 umwelt.info – Metadaten-Index – Option: Sprachmodell (White Box)

Siehe auch (KI-Komponenten):

- 2.2.1, Recommender System (RS) – Merklisten und Collaborative Filtering
- 2.2.2, KI-Suche
- 2.2.3, Linked Data (LD) Service
- 2.2.4, Search Prediction
- 2.2.5, Search RS
- 2.2.6, Term RS
- 2.2.7, Q/A Retrieval
- 2.2.8, KI-MQA

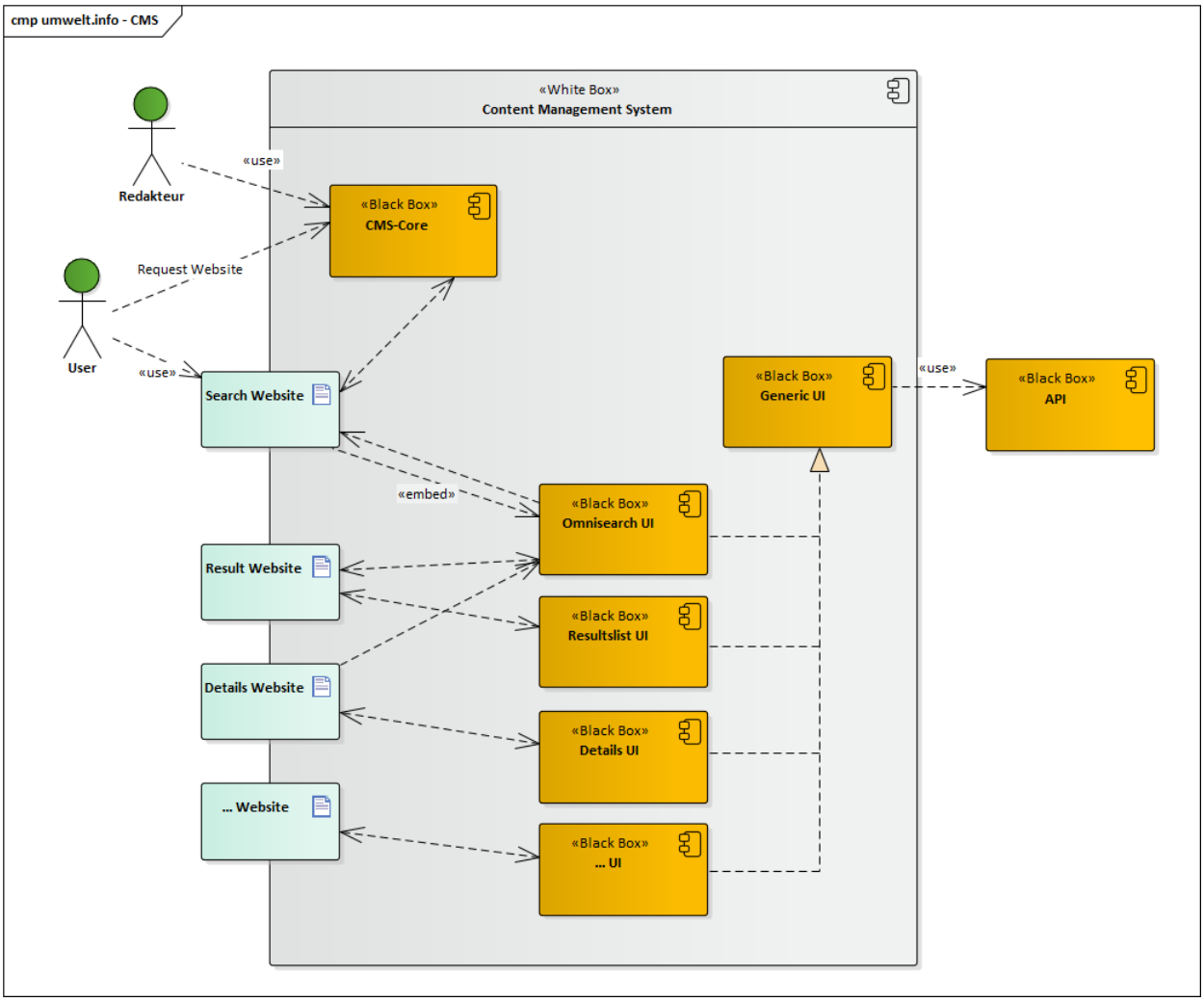
Schnittstelle(n)

Da es sich hier im Wesentlichen um eine Aggregation der in Ebene 2 beschriebenen spezialisierten Metadaten-Index Komponenten Linked Data und Sprachmodell handelt, werden exakt die Schnittstellen bereitgestellt und benötigt wie in diesen beiden Komponenten beschrieben. Spezielle Funktionen und Schnittstellen werden durch die einzelnen KI-Komponenten eingeführt.

2.3 Ebene 2 umwelt.info – Content Management System (White Box)

Das Content Management System, das in der folgenden Abbildung 4 dargestellt ist, stellt die eigentliche Portal Benutzerschnittstelle. Es liefert die Websites für die Nutzenden aus.

Abbildung 4: Ebene 2 umwelt.info – Content Management System (White Box)



Quelle: eigene Darstellung, con terra GmbH

2.3.1 CMS Core

Beschreibung

Die CMS Core-Komponente stellt den Nutzer*innen Websites bereit. Sie nimmt die Seitenaufrufe entgegen, rendert die angefragten Websites und sendet sie an den anfragenden Client (Browser) der Nutzer*innen.

Redakteur*innen können Websites erstellen und bearbeiten. Dabei muss das CMS über einen Mechanismus ermöglichen, die UI-Komponenten (z. B. Suche, Benachrichtigungen, etc.) in die Websites einzubinden.

Weitere Details

Die CMS Core-Komponente speichert die Inhalte der Seiten des umwelt.info Portals und rendert diese für die Auslieferung an die Nutzer*innen, wenn diese eine Portalseite per http(s) mit ihren Webbrowsern anfragen. Die Websites des umwelt.info Portals integrieren die UI-Komponenten des umwelt.info Portals. Dabei sind einige Funktionen (z. B. die Suche) für alle Nutzer*innen verfügbar und andere nur für angemeldete bzw. berechtigte Nutzer*innen (z. B. Personalisierte Favoriten).

Websites binden JavaScript-Code der UI-Komponenten ein, die von den UI-Komponenten eigenständig oder über das CMS bereitgestellt werden.

Schnittstelle(n)

Bereitgestellt:

- Websites des umwelt.info Portals

Qualitäts-/Leistungsmerkmale

Besonders hohe Verfügbarkeit und hohe Skalierbarkeit da es sich um eine der zentralen Komponenten des Systems handelt

Offene Punkte/Probleme/Risiken

- Es soll eine möglichst lose Kopplung der UI-Komponenten mit dem CMS Core angestrebt werden.
- Risiken und Offene Punkte werden in Kapitel 10.1, Einsatz des Government Site Builder als Content Management System beschrieben.

2.3.2 User Interfaces (UI) Components

Beschreibung

UI-Komponenten sind z. B. die in Abbildung 3 dargestellten „Suchschlitz“, „Ergebnisseite“ oder „Detailseite“. Diese Komponenten sind die Softwareteile, die die Funktionen des umwelt.info Portals für die Nutzer*innen in intuitiver und interaktiver Form nutzbar machen. Die Benutzeroberfläche des umwelt.info Portals setzt sich aus einer oder mehreren Websites zusammen, die dabei die UI-Komponenten einbinden.

Weitere Details

Da die Umsetzung der Benutzeroberfläche als Web-Portal geplant ist, müssen UI-Komponenten im Browser der Nutzer*innen laufen. Interaktive Elemente sollten daher (gemäß Stand der Technik) mittels JavaScript/HTML5 implementiert werden. Dabei muss in der Implementierungsphase entschieden werden, welche Elemente einer UI-Komponente mittels HTML dargestellt (serverseitig gerendert) und welche Elemente durch DOM-Manipulation (clientseitig, mittels JavaScript im Browser) dynamisch verändert werden. Letzteres betrifft etwa Interaktionen der Nutzer*innen mit den UI-Komponenten oder das Nachladen von Inhalten, ohne Neuladen einer Webseite.

UI-Komponenten können außerdem miteinander interagieren, abstrakt sein (nur Funktionalität bereitstellen) und andere UI-Komponenten enthalten.

Die UI-Komponenten werden im Designkonzept (unveröffentlichtes Projektdokument des Umweltbundesamtes) beschrieben. Einige UI-Komponenten sind zum Beispiel:

- Suchschlitz

- Suchergebnisseite
- Detailseite
- Suche Erweitern (vgl. Kapitel 4.3 (Anwendungsmöglichkeiten von Linked Data und KI-Methoden – Suche Erweitern) im Umsetzungskonzept)
- Mashup / SPARQL (vgl. Kapitel 4.6 (Anwendungsmöglichkeiten von Linked Data und KI-Methoden – Erweiterte Antworten / Mashups) und Kapitel 4.9 (Anwendungsmöglichkeiten von Linked Data und KI-Methoden – SPARQL-Client) im Umsetzungskonzept)

Schnittstelle(n)

Benötigt:

- API

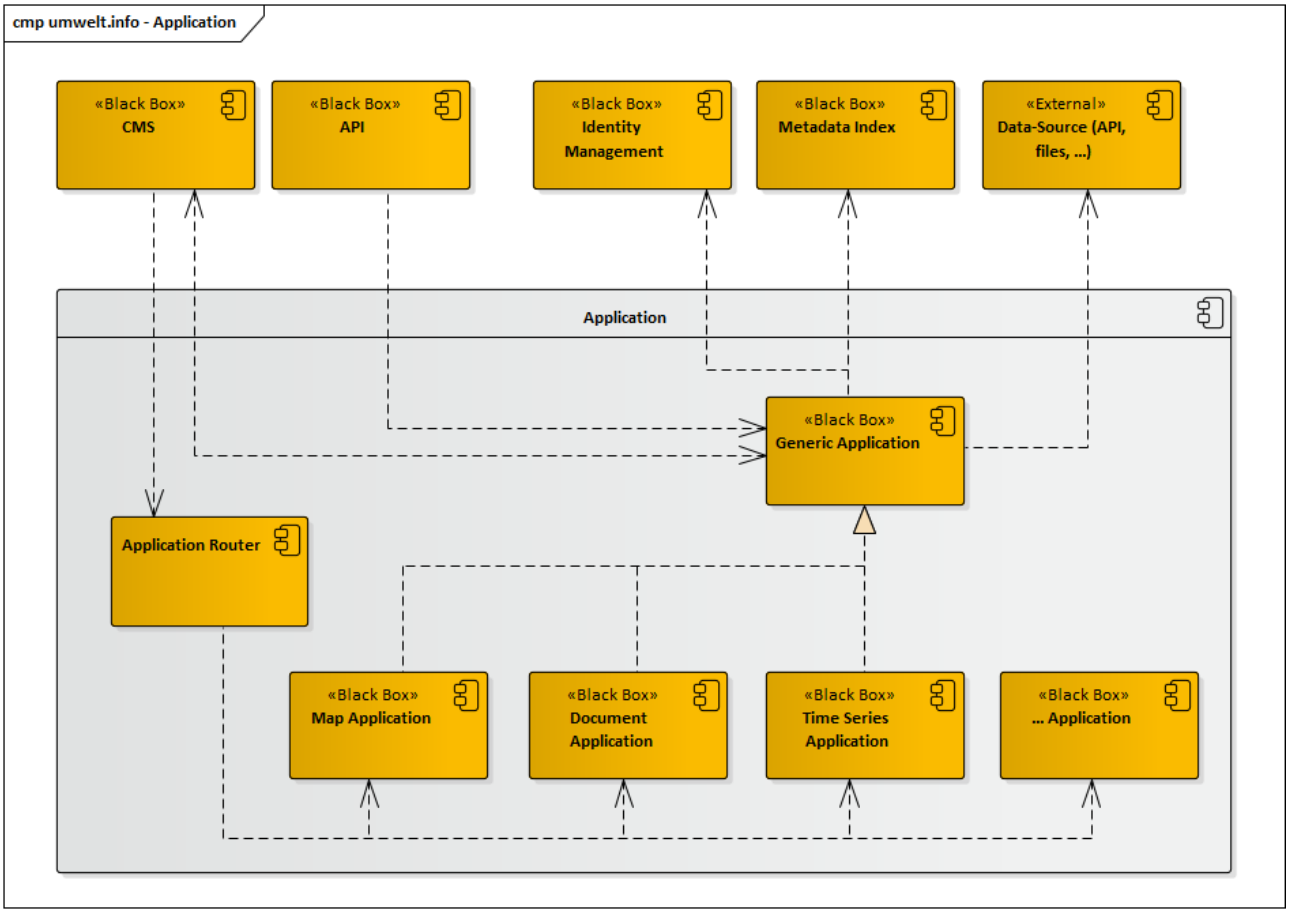
Qualitäts-/Leistungsmerkmale

Es soll eine möglichst lose Kopplung der UI-Komponenten untereinander, sowie mit dem CMS Core angestrebt werden, um eine hohe Wart- und Testbarkeit der UI-Komponenten zu ermöglichen (vgl. 5.3.2).

2.4 Ebene 2 umwelt.info – Application (White Box)

Die Application-Komponente stellt den Nutzer*innen unterschiedliche (passende) Applications bereit, um die Daten im umwelt.info System miteinander zu kombinieren und zu visualisieren. Dafür ist umwelt.info offen für Erweiterungen mit verschiedenen (unabhängigen) Anwendungen. Damit könnte umwelt.info die aus unterschiedlichen externen Quellen stammenden unbekannten Daten, mittels verschiedener Funktionen verarbeiten. Für bestimmte Standardformate kann dabei initial eine spezielle Application-Komponente vorgesehen werden.

Abbildung 5: Ebene 2 umwelt.info – Application (White Box)



Quelle: eigene Darstellung, con terra GmbH

2.4.1 Application Router

Beschreibung

Der Application Router ist die Komponente, die bei einer Anfrage durch die Nutzer*innen für einen bestimmten Datentyp oder eine bestimmte Quelle an eine bestimmte Anwendung weiterleitet.

Weitere Details

Der Application Router wird durch parametrisierte Verlinkungen innerhalb der Webseiten des CMS angesteuert. Durch die Weiterleitung der Verlinkungen wird die passende Anwendung angesteuert. Die Weiterleitung erfolgt anhand von Parametern in der URL, die explizit eine bestimmte Anwendung steuern oder implizit durch die Parameter der Datenquelle und Datentypen gesteuert werden.

Schnittstelle(n)

Benötigt:

- Zugriff auf die verschiedenen Anwendungen in der Application

Qualitäts-/Leistungsmerkmale

Kopplung zwischen CMS und den verschiedenen Anwendungen, Weiterleitung an die Anwendungen

Offene Punkte/Probleme/Risiken

Es ist möglich, dass es für bestimmte Datenquellen keine geeignete Anwendung integriert ist.

2.4.2 Map/Document/Time Series Application

Beschreibung

Über die verschiedenen Anwendungen Map, Document und Time Series haben die Nutzer*innen die Möglichkeit sich die Daten visuell darstellen zu lassen.

Weitere Details

In der Map Application werden den Nutzer*innen verschiedene Werkzeuge, wie z. B. eine Karte, Navigationswerkzeuge, Analysewerkzeuge, Koordinatenanzeige, FeatureInfo, Table of Content, Maßstabsanzeige und Legende bereitgestellt.

Die Document Application verfügt über z. B. Navigationswerkzeuge und Analysewerkzeuge und bei der Time Series Application werden den Nutzer*innen z. B. Navigationswerkzeuge, Analysewerkzeuge sowie Diagrammtypen bereitgestellt.

Mit Hilfe der verschiedenen Anwendungen und deren Werkzeuge, können die Nutzer*innen weitere Erkenntnisse aus den Daten ableiten.

Schnittstelle(n)

Bereitgestellt:

- Eigene API (optional)

Benötigt:

- API
- Identity Management

Offene Punkte/Probleme/Risiken

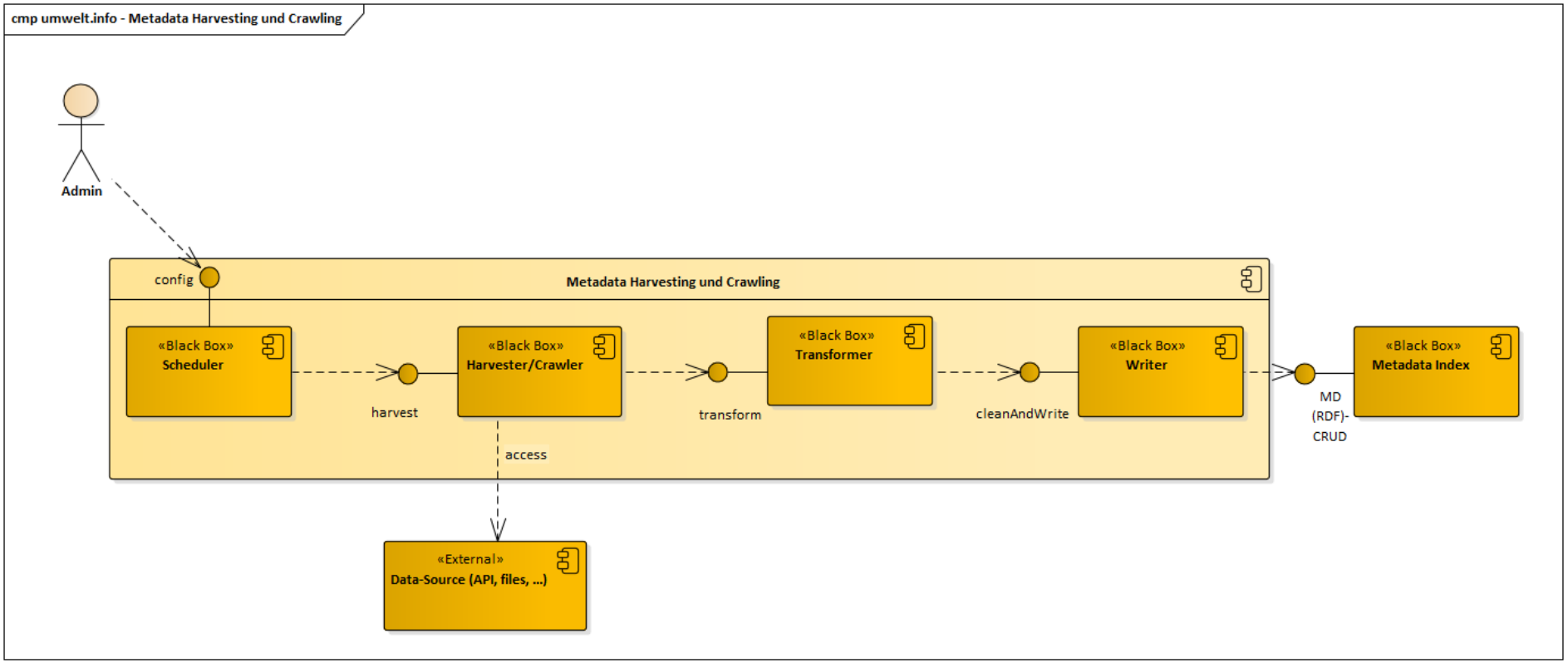
Offen: Optionale Anbindung des Data Cube

2.5 Ebene 2 umwelt.info – Metadata Harvesting und Crawling (White Box)

Das primäre Ziel dieser Komponente besteht darin, Metadaten regelmäßig aus den verschiedenen Datenquellen außerhalb des Systems zu ermitteln, in das interne Metadatenmodell zu überführen und die Metadaten dann zu indizieren und somit auffindbar zu machen.

Der Ansatz des Metadata Harvesting und Crawling folgt dem Extract-Transform-Load-Prinzip (einige Ideen der im Folgenden dargestellten Harvesting und Crawling Architektur entstammen [13]): Extrahieren (der Metadaten aus den verschiedenen Quellen), Transformieren (der Metadaten in das Zielformat) und Laden (des Zielformates in den Metadaten-Index). Die einzelnen involvierten Services besitzen jeweils eine Web-Schnittstelle über die sie verbunden und orchestriert werden können. Es gibt keine zentrale Instanz, die für die Orchestrierung der Services zuständig ist. Vielmehr wird jedem Service über eine ihm zugeführte Prozessbeschreibung mitgeteilt, was er tun soll und welchen Service er als nächstes aufzurufen hat (an den er dann auch die Servicebeschreibung weiterleitet):

Abbildung 6: Ebene 2 umwelt.info – Metadata Harvesting und Crawling (White Box)



Quelle: eigene Darstellung, con terra GmbH

2.5.1 Scheduler

Beschreibung

Der Scheduler managt, wann für eine Datenquelle der Metadaten-Ermittlungsprozess zu starten ist und wählt die für die Datenquelle richtige Harvesting-Pipeline, indem der passende Harvester bzw. Crawler aufgerufen und ihm den Harvesting/Crawling Auftrag übergibt.

Schnittstelle(n)

Bereitgestellt:

- config ist eine Schnittstelle, die es erlaubt, den Scheduler zu konfigurieren.

Benötigt:

- REST-Schnittstelle zum Initiieren einer Harvesting/Crawling Pipeline durch Übergabe eines Harvesting/Crawling Auftrages an einen Harvester bzw. Crawler.

Qualitäts-/Leistungsmerkmale

- Hohe Verfügbarkeit,
- Gute Logging- und Monitoringfähigkeit da der Scheduler die zentrale Instanz für das Harvesting darstellt

2.5.2 Harvester / Crawler

Beschreibung

Der erste Schritt im Harvesting-Prozess, das eigentliche Abholen der Metadaten, wird von sogenannten Metadata-Harvestern/Crawlern übernommen. Es gibt verschiedene Typen von Metadaten Harvester bzw. Crawler, die jeweils für das Harvesten eines bestimmten Typs von Datenquelle (mit bestimmter Schnittstelle (Interface)) zuständig sind. Je mehr Datenquellen geharvestet werden sollen, umso mehr Harvester bzw. Crawler müssen bereitgestellt werden (sofern sich die Schnittstellen der Datenquellen unterscheiden).

Schnittstelle(n)

Bereitgestellt:

- Die Metadaten Harvester bzw. Crawler sollen eine Schnittstelle bereitstellen, mit der sie von außen einen Harvesting/Crawling Auftrag entgegennehmen

Benötigt:

- Schnittstelle der Datenquelle für die der Harvester bzw. Crawler konfiguriert ist
- REST-Schnittstelle eines Transformers, dem die ermittelten, (Meta)Daten sowie der Harvesting/Crawling-Auftrag einer Harvesting/Crawling Iteration übergeben werden.

Qualitäts-/Leistungsmerkmale

- Verschiedene Harvester bzw. Crawler (auch des gleichen Typs) müssen parallel ablauffähig sein.

- Die Harvester bzw. Crawler sollten durch die Portalbetreiber konfigurierbar sein, so dass etwa die Anfrage an eine zu harvestende Datenquelle konfiguriert werden kann.
- Die Schnittstellen sollten idealerweise die Möglichkeit besitzen, lediglich die Daten/Metadaten anfordern zu können, die sich seit dem letzten „Harvesting" geändert haben, um nicht immer alle Daten/Metadaten der Daten-Ressource wiederholt harvesten zu müssen.
- Die Harvester bzw. Crawler müssen die Daten so an einen Transformer übergeben, wie er sie erwartet, damit er diese in das Ziel-Metadatenformat umwandeln kann

2.5.3 Transformer

Beschreibung

Eine wesentliche Aufgabe ist das Überführen („mappen") der ermittelten Metadaten in das gemeinsame (im Metadaten-Index verwendete) Metadaten Schema (vgl. Kapitel 5.1.1). Der Transformer konvertiert entweder direkt (im Falle, dass die Datenquelle bereits Metadaten liefert) in das interne Metadatenmodell und übergibt die resultierenden Metadaten an einen Writer zur Speicherung oder es werden (falls die Datenquelle zwar Daten aber keine Metadaten liefert) Metadaten entsprechend dem internen Metadatenmodell zunächst aus den Daten automatisiert abgeleitet und dann an den Writer übergeben.

Die Transformer übernehmen auch das „semantische Anreichern" der Metadaten. Hier sollte etwa der Semantische Netzwerk Service (SNS) [3] zum Einsatz kommen, zur Verbesserung der Metadaten durch das „Verlinken" mit Begrifflichkeiten des Umwelthesaurus (UMTHES) oder um für unstrukturierte Daten SNS Begriffe entsprechend UMTHES herauszuziehen und zu linken (s. 2.1.2). Die Transformer sollten so arbeiten, dass sie zunächst die evidenten Texte (description, title, keywords etc,) aus den geharvesteten Metadaten ermitteln und in den SNS geben und erst dann die finalen DCAT-AP.de bzw. Schema.org (s. Kap. 5.1.1) Dokumente (inkl. der UMTHES Schlagworte) erzeugen.

Schnittstelle(n)

Bereitgestellt:

- Die Transformer müssen eine Schnittstelle bereitstellen, die geharvestete (Meta-)Daten (und evtl. weitere Anweisungen) entgegennimmt.

Benötigt:

- REST-Schnittstelle eines Writers, der einen Satz Metadaten (und evtl. weitere Anweisungen) entgegennimmt.

Qualitäts-/Leistungsmerkmale

- Hohe Verfügbarkeit
- Verschiedene Transformer (auch des gleichen Typs) müssen parallel ablauffähig sein.
- Die Transformer sollten durch die Portalbetreiber konfigurierbar sein, so dass etwa das mapping (etwa von keywords auf Thesauri/Ontologien) verändert werden kann.

2.5.4 Writer

Beschreibung

Der Writer schiebt die bereits im passenden Metadatenmodell vorliegenden und semantisch angereicherten Metadaten in den Index, wo sie auf der Basis bestimmter Elemente indiziert und gespeichert werden. Vom Metadaten-Index sollte für die ermittelten Dokumente zusätzlich eine Volltextindizierung veranlasst werden.

Für unstrukturierte Daten und Informationen kann die Volltextindizierung die einzige Möglichkeit der Indizierung sein, falls sich von den Harvestern und Crawlern keine Metadaten extrahieren lassen konnten, die das interne Metadatenmodell befriedigen.

Der Writer veranlasst auch gewisse Bereinigungen der Daten im Metadaten-Index, z. B. können nicht (mehr) existierende Metadaten gelöscht werden. Gelöschte Metadaten könnten optional in ein Archiv überführt werden, um später etwa nachvollziehen zu können, welche Daten mal verfügbar waren, mittlerweile aber nicht mehr sind. Das Ganze kann auf unterschiedliche Weisen implementiert werden. Eine Möglichkeit besteht etwa darin, dass beim ersten Schreiben eines „Harvesting-Zyklus“ für die jeweilige Datenquelle der im Index verfügbare Metadatenbestand dieser Quelle in ein Archiv überführt und dieser aus dem Index entfernt wird. Am Ende des „Harvesting-Zyklus“ (erkennbar etwa daran, dass die von einer Datenquelle mit entsprechender Schnittstelle gelieferte Anzahl Metadaten (das harvesting geschieht mittels Anfrage von Blöcken, aka “paging“) kleiner ist als die angefragte Anzahl) wird dann ein Abgleich der im aktualisierten Index verfügbaren Metadaten mit denen im Archiv gemacht und es verbleiben nur die im Archiv die nicht im Index sind.

Ein weiteres Problem stellen Duplikate von Metadaten dar. Diese können etwa dadurch entstehen, dass verschiedene Metadatenquellen ihrerseits bereits Metadatenquellen geharvestet haben, welche auch von umwelt.info direkt geharvestet werden. Da die Metadaten üblicherweise über eindeutige Identifier verfügen (sollten) (so z. B. „fileIdentifier“ in INSPIRE [5]) sollte nur die jeweils eine (aktuelle) Version im Index verbleiben. Üblicherweise könnten solche Redundanzen verhindert werden, indem Metadatenquellen nur nach ihren originären Metadaten (und nicht den geharvesteten) abgefragt werden. Da aber einige Schnittstellen (Protokolle) dieses nicht unterstützen oder Implementierungen sich nicht korrekt an die Vorgaben des Protokolls halten (obwohl dieses das vorsieht (z. B. bei OGC CSW)), gelangen geharvestete Metadaten in den Index und können so redundant vorkommen. Der Writer (bzw. eine weitere separate Komponente) versucht, solche Redundanzen zu erkennen und veranlasst nur die jeweils aktuellere Version (erkennbar am „datestamp“ der Metadaten) im Index zu speichern.

Weitere Fälle sind kompliziert oder nur schwer auflösbar. Solche problematischen Doppelungen entstehen, wenn unterschiedliche Metadaten (mit einem unterschiedlichen Identifier) in den Index gelangen die entweder dieselbe Datenquelle beschreiben (der Datenzugriffslink also prinzipiell identisch ist) oder die Metadaten jeweils über einen anderen Datenzugriffslink auf die Daten verweisen. Im ersten Fall könnte man z. B. einen Hash für den normalisierten kanonischen Datenzugriff errechnen und bei Gleichheit nur den jeweils aktuelleren Metadatensatz indizieren. Im anderen Fall wird es noch schwieriger, hier müsste anhand des Inhaltes von zwei Metadatensätzen ermittelt werden, ob beide den gleichen Datensatz beschreiben. Hierzu müsste ein Algorithmus zur Erkennung entwickelt oder ein bereits anderswo (z. B. beim BKG) etablierter verwendet werden oder ggf. mit redaktionellen Maßnahmen gegengesteuert werden.

Schnittstelle(n)

Bereitgestellt:

- Die Writer müssen eine Schnittstelle bereitstellen, die einen Satz Metadaten und weitere Anweisungen entgegennimmt.

Benötigt:

- MD (RDF) CRUD-Schnittstelle eines Metadaten-Index, der einen Satz Metadaten und weitere Anweisungen entgegennimmt.

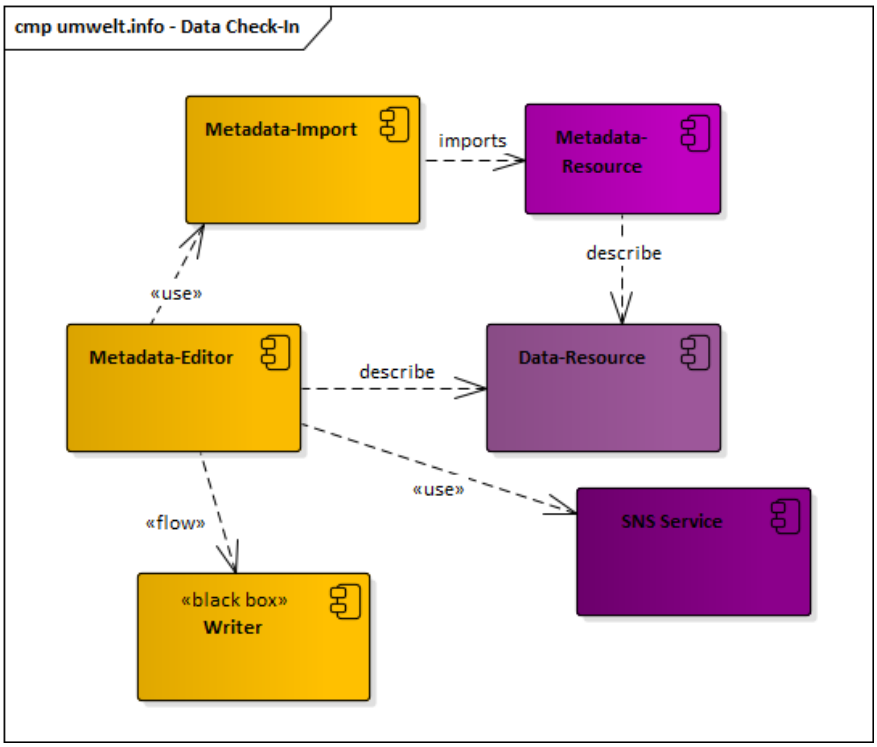
Qualitäts-/Leistungsmerkmale

- Hohe Verfügbarkeit
- Verschiedene Writer (auch des gleichen Typs) müssen parallel ablauffähig sein.

2.6 Ebene 2 umwelt.info – Data Check-In (White Box)

Zur Erfassung von Metadaten ist ein Editor entsprechend dem internen Metadatenmodell bereitzustellen, der neben den Standardmetadaten auch die Zugriffsinformationen auf das Objekt erfasst. Alternativ ist auch der Import bestehender Metadaten für das Objekt mittels der Metadata-Import Komponente möglich.

Abbildung 7: Ebene 2 umwelt.info – Data Check-In (White Box)



Quelle: eigene Darstellung, con terra GmbH

2.6.1 Metadata-Editor

Beschreibung

Diese Komponente ermöglicht die manuelle Erfassung von Metadaten zu einer Data-Source.

Anstelle einer komplett neuen Erfassung der Metadaten können auch bereits existierende Metadaten via Import in den Metadata-Editor geladen werden.

Nach Abschluss der Bearbeitung übergibt der Metadata-Editor die Metadaten entsprechend dem internen Metadatenformat über einen Writer an den Metadaten-Index.

Weitere Details

Es lässt sich direkt der SNS-Service [3] einbinden, der es ermöglicht z. B. einfache Keywords auf Elemente eines existierenden Vokabulars oder einer Ontologie zu mappen, welche dem originären Wert semantisch entsprechen. Ein Beispiel ist etwa das „Verlinken" auf ein INSPIRE Annex Thema "Uniform Resource Identifier" (URI) des INSPIRE GEMET Thesaurus.

Schnittstelle(n)

Benötigt:

- REST-Schnittstelle eines Writers, der einen Satz Metadaten (und evtl. weitere Anweisungen) entgegennimmt.
- Import Schnittstelle der Metadata-Import Komponente

Qualitäts-/Leistungsmerkmale

Der Metadata-Editor sollte eine UI besitzen mit der möglichst einfach und intuitiv Metadaten erfasst werden können. Der Editor sollte nach den Prinzipien des User Experience Design (vgl. umwelt.info Designkonzept) konzipiert werden.

Offene Punkte/Probleme/Risiken

Es ist zu klären, ob der Import die Metadaten grundsätzlich zunächst in den Editor laden soll oder ob auch ein direkter Import in den Metadaten-Index vorzusehen ist. Hier würden die gleichen QA-Maßnahmen greifen, wie beim Harvesten/Crawlen.

2.6.2 Metadata-Import

Beschreibung

Hiermit können bereits existierende Metadaten via Import in den Metadata-Editor geladen werden.

Weitere Details

Die Komponente zum Metadata-Import der Metadaten zu einer Data-Source, kann bereits im internen Metadatenformat vorliegende Metadaten (z. B. in XML/RDF oder JSON-LD) in den Editor laden oder diese direkt bis zum Metadaten-Index durchleiten.

Schnittstelle(n)

Bereitgestellt:

- Import Schnittstelle

Benötigt:

- Zugriff auf das lokale Verzeichnissystem

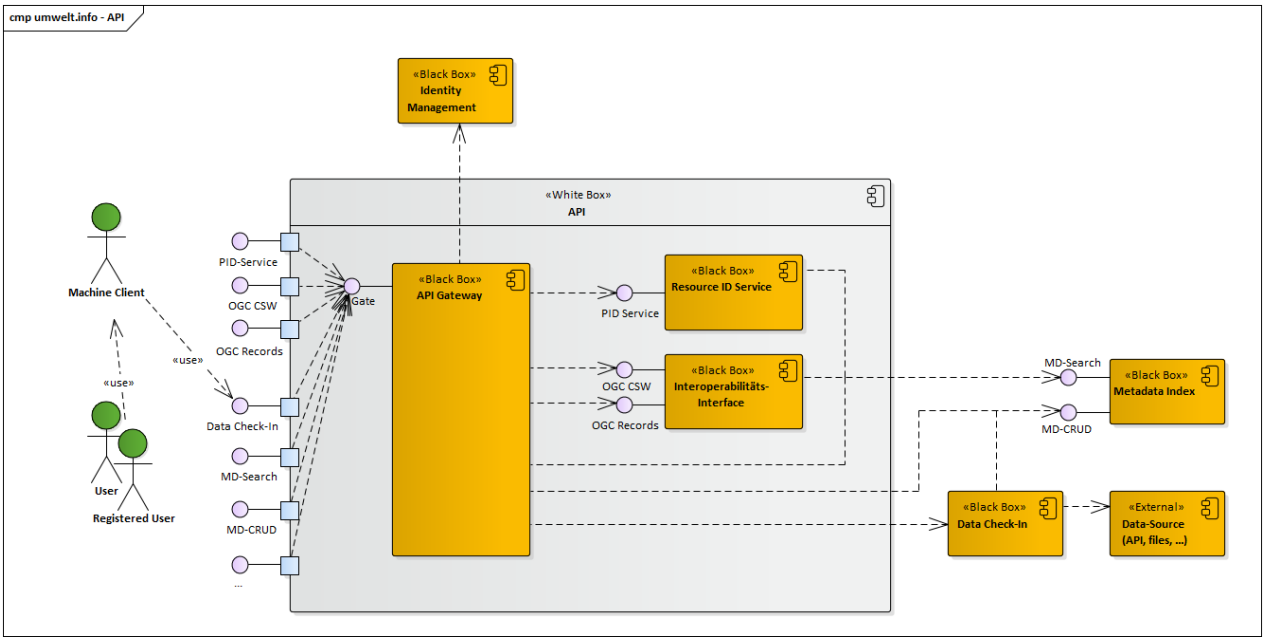
Qualitäts-/Leistungsmerkmale

Import der Metadaten

2.7 Ebene 2 umwelt.info – API (White Box)

Die Funktionen des umwelt.info Portals sind über die API auch für andere, externe Anwendungen nutzbar. Dazu werden über das API-Gateway die benötigten REST-Schnittstellen des Data Check-In und des Metadaten-Index, entsprechend den Berechtigungen der Nutzer*innen zugänglich gemacht. Darüber hinaus stellt eine weitere Komponente eine Schnittstelle für die Interoperabilität mit anderen Portalen bereit.

Abbildung 8: Ebene 2 umwelt.info – API (White Box)



Quelle: eigene Darstellung, con terra GmbH

2.7.1 API-Gateway

Beschreibung

Nur über das API-Gateway können externe Anwendungen auf die APIs des Portals zugreifen. Dabei kann das API-Gateway so konfiguriert werden, dass nur bestimmte Funktionen der APIs öffentlich und andere Funktionen nur für autorisierte Clients zugänglich sind.

Weitere Details

Auch alle von der UI benötigten Funktionen des Portals sind ausschließlich über das API-Gateway verfügbar. Nach Möglichkeit sollte auch das CMS und die damit verbundene Auslieferung der HTML-Websites des Portals nur über das API-Gateway möglich sein.

Schnittstelle(n)

Das API-Gateway steuert den Zugriff auf die APIs aller internen Komponenten. Neben den APIs sind für den Nutzer folgende Komponenten nur über das API-Gateway erreichbar:

- Data Check-In
- Identity Management
- Metadaten-Index (MD-Search, MD-CRUD)

Benötigt:

- Zu veröffentlichende Schnittstellen aller internen Komponenten (Metadaten-Index, Data Check-In, Identity Management, ... <weitere>)
- Identity Management

Qualitäts-/Leistungsmerkmale

Für die APIs wird eine hohe Sicherheit, Verfügbarkeit und Skalierbarkeit benötigt. Für alle APIs die für externe Anwendungen verfügbar gemacht werden sollen (z. B. für die UIs oder Skripte zum Datenbereitstellen), ist das Gateway entsprechend zu konfigurieren.

2.7.2 Interoperabilitätsinterface

Beschreibung

Diese Komponente stellt Interoperabilitätsinterfaces (APIs) zur Verfügung.

Weitere Details

Für einen interoperable Zugriff auf die Metadaten von umwelt.info (etwa im Rahmen einer Suche oder eines Harvesting durch andere Metadatenportalen oder Suchmaschinen) wird eine OGC CSW AP ISO (INSPIRE)- und eine OGC API Records Schnittstelle bereitgestellt.

Schnittstelle(n)

Bereitgestellt:

- OGC CSW ISO (inkl. INSPIRE) (optional)
- OGC API Records

Benötigt:

- MD-Search des Metadaten-Index

Qualitäts-/Leistungsmerkmale

Einhalten der Spezifikationen der Interoperabilitätsinterfaces

2.7.3 Resource Identification Service

Beschreibung

Der Resource Identification Service (Resource ID Service) ermöglicht die Abfrage von Metadaten als HTML und JSON, sowie ggf. weiterer Formate, unter Angabe einer ID.

Weitere Details

Als Ergebnis erhält die anfragende Komponente (oder Nutzer*innen) eine eindeutige URL, die vom System nicht verändert wird (Permalink). Diese eindeutigen URLs werden zum Beispiel in den Favoriten der Nutzer*innen verwendet.

Schnittstelle(n)

Bereitgestellt:

- PID Service (Persistent Identifier Service)

Benötigt werden:

- MD-Search des Metadaten-Index

Qualitäts-/Leistungsmerkmale

- Hohe Sicherheit
- Hohe Verfügbarkeit

Offene Punkte/Probleme/Risiken

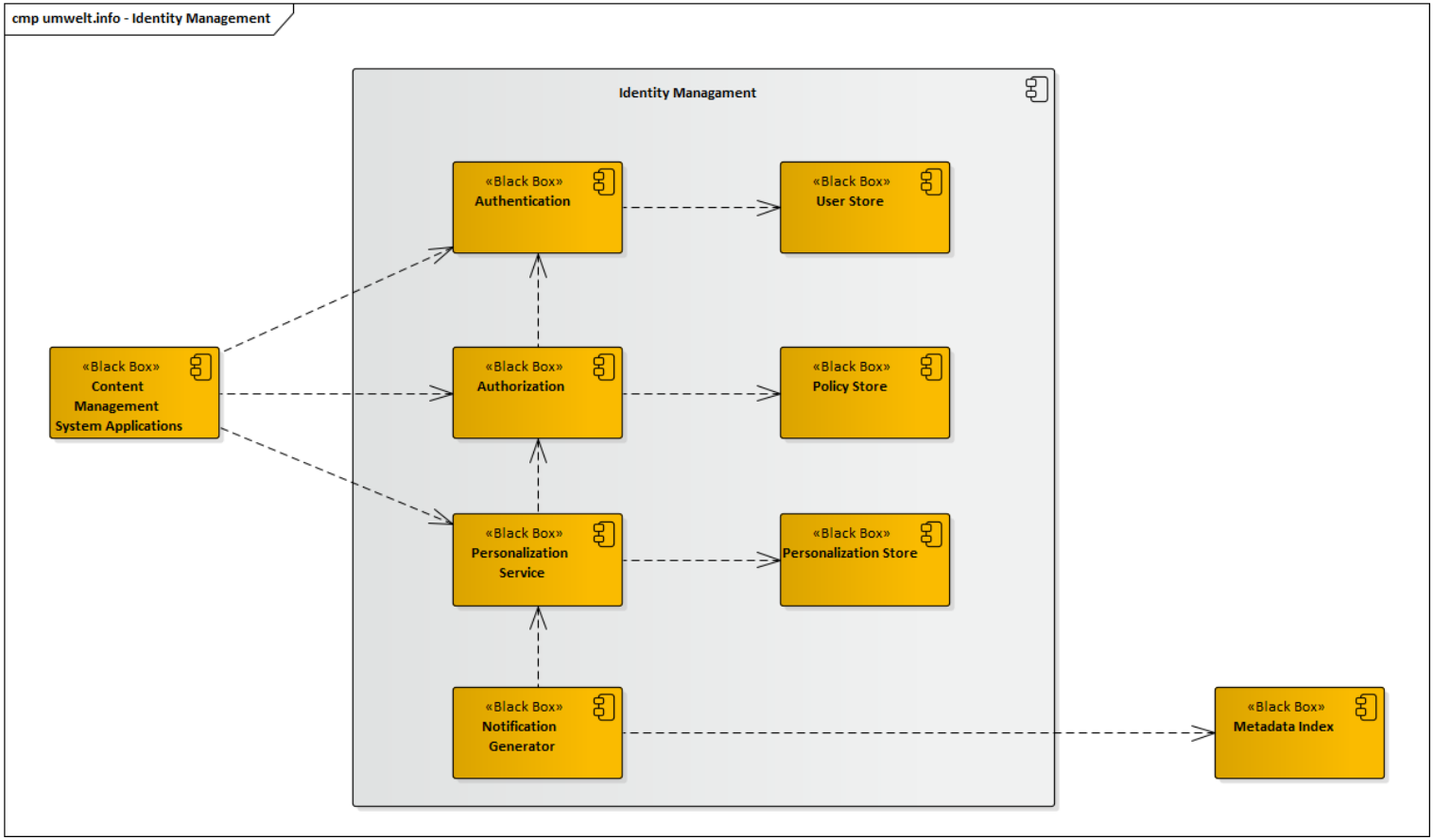
Es ist unklar, wie weit die Garantien für die Permanenz gehen, d.h. welche Änderungen in den Datenquellen hier abgedeckt sind (also insbesondere ob die Permalinks hier aus Permalinks der Quellen abgeleitet werden).

Zusätzlich ist nicht klar, wo die ggf. notwendige Persistenz für diese Permalinks angesiedelt ist, direkt in diesem Dienst oder innerhalb des Metadaten-Index.

2.8 Ebene 2 umwelt.info – Identity Management (White Box)

Das Identity Management ist für alle Funktionen im umwelt.info Portal relevant, die über Schnittstellen nach Außen verfügbar sind und die nicht von einem anonymen Nutzer durchgeführt werden dürfen. Zu den grundlegenden Funktionen gehören An- und Abmeldung, Zugriffskontrolle und die Breitstellung von nutzerspezifischen Informationen.

Abbildung 9: Ebene 2 umwelt.info - Identity Management (White Box)



Quelle: eigene Darstellung, con terra GmbH

2.8.1 Authentication

Beschreibung

Die Authentication ermöglicht die Anmeldung am umwelt.info Portal. Die Authentication wird von allen Clients verwendet, um eine Single Sign-On Session mit dem umwelt.info Portal herzustellen. Dafür stellt die Authentication eine Anmeldeseite bereit. Nach erfolgreicher Anmeldung erhält der Client von der Authentication Komponente ein Security Token, das die Identität der angemeldeten Nutzer*innen bestätigt.

Intern verwendet die Authentication einen User Store, in dem alle erforderlichen Informationen über die Nutzer*innen abgelegt sind.

Weitere Details

Für die Authentication gibt es bereits eine Auswahl von Standardprodukten, die die notwendigen “Single Sign-On" (SSO) Protokolle unterstützen. Ein solches Produkt sollte verwendet und für umwelt.info entsprechend angepasst werden.

Um den Zugriff auf geschützte APIs durch nicht-UI Clients (z. B. durch Skripting) zu ermöglichen, kann die Authentication auch dafür geeignete Schnittstellen unterstützen. Dieser Anwendungsfall kann aber auch alternativ gelöst werden, z. B. durch statische API-Tokens.

Schnittstelle(n)

Bereitgestellt:

- Login: Kann von Clients wie den UI-Komponenten verwendet werden, um Nutzer*innen am Portal anzumelden. Dabei wird eine SSO-Session erzeugt, so dass Nutzer*innen sich nur einmal anmelden müssen. Hierfür sollten bereits gebräuchliche Standardprotokolle wie OpenID Connect, CAS-Protokoll oder SAML Web Browser Profile verwendet werden.
- UI für die Anmeldung.
- Logout: Kann von Clients verwendet werden, um die SSO-Session des umwelt.info Portals zu beenden.

Benötigt:

- Schnittstelle des User Store: Über diese Schnittstelle werden die Informationen der Nutzer*innen abgefragt.

Qualitäts-/Leistungsmerkmale

- Hohe Sicherheit.
- Hohe Verfügbarkeit
- Unterstützung von Standardprotokollen, um die Integration von anderen Komponenten zu erleichtern.

Offene Punkte/Probleme/Risiken

Risiken:

- Je nach Technologie der anderen Portalkomponenten ist die Unterstützung bestimmter Protokolle vorausgesetzt (z. B. bringt der Government Site Builder mit CAS eine eigene SSO-Lösung mit).
- Im Rahmen der Umsetzung sollte beachtet werden, dass die Personalisierungsfunktionen nicht stark mit den übrigen Funktionen des Identity Managements gekoppelt werden müssen. Es kann evtl. eine Unterteilung in mehrere Komponenten erfolgen.

2.8.2 Authorization

Beschreibung

Ermöglicht die Zugriffssteuerung auf Funktionen im umwelt.info Portal.

Alle Anfragen von Clients durchlaufen zunächst die Autorisierung im API-Gateway, bevor sie zu den entsprechenden Backend-Diensten weitergeleitet werden. Dies ist im White Box Diagramm für das Identity Management durch Abhängigkeiten der dargestellten Komponenten Personalization Service und Content Management System Applications verdeutlicht.

Die Autorisierung erfolgt anhand eines Security Tokens, das in der Regel durch die Authentication Komponente erzeugt wurde. Unterstützt werden müssen sowohl Browser-basierte Clients als auch maschinelle Clients (z. B. Skripting)

Die Authorization übernimmt die Zugriffsentscheidung anhand von Regeln, die in einem Policy Store abgelegt sind. Dabei werden die erforderlichen Rechte mit den Eigenschaften der Nutzer*innen (z. B. Rollen) abgeglichen, die anhand des Security Tokens ermittelt werden können.

Die Anfrage, die an den Backend-Dienst weitergeleitet wird, enthält anstelle des Security Tokens ein Token mit den Nutzerattributen, z. B. in Form eines JSON Web Token. Dieses Token kann z. B. von der Authentifizierung bereitgestellt werden. Alternativ kann dafür auch direkt auf den User Store zugegriffen werden. Anhand der Attribute kann der Backend-Dienst weitere Funktionalitäten implementieren oder Zugriffsentscheidungen treffen. Ein Beispiel dafür ist die Rückgabe der gespeicherten Suchen, bei denen über die Nutzer-ID gefiltert werden muss.

Weitere Details

In modernen Microservice Architekturen wird die Autorisierung in der Regel von einem zentralen API-Gateway übernommen. Feingranulare Zugriffsentscheidungen lassen sich in einigen Fällen aber effizienter direkt in den einzelnen Portalkomponenten realisieren.

Schnittstelle(n)

Bereitgestellt:

- Die Autorisierung muss die Schnittstelle des Backend-Dienstes anbieten. Es wird eine Anfrage mit den Attributen und Rollen des Nutzerprofils an den Backend Dienst weitergeleitet.

Benötigt:

- Schnittstelle, über die aus einem Security Token die Nutzerattribute ermittelt werden können.
- Schnittstelle zum Policy Store.
- Schnittstelle des Backend-Dienstes.

Qualitäts-/Leistungsmerkmale

- Hohe Sicherheit.
- Hohe Verfügbarkeit

Offene Punkte/Probleme/Risiken

Je nach Technologie der anderen Teilkomponenten ist die Autorisierung bereits implementiert (z. B. verwenden die Komponenten des Government Site Builder teilweise lokale Autorisierung mit Spring Security).

2.8.3 User Store

Beschreibung

Ermöglicht das Speichern aller Informationen, die für Nutzer*innen des umwelt.info Portals erforderlich sind. Der User Store enthält die Profilinformationen für alle Nutzer*innen, einschließlich der verschlüsselten Passwörter. Außerdem werden hier alle Informationen abgelegt, die für die Autorisierung erforderlich sind, wie beispielsweise Rollen.

Weitere Details

In der Regel wird der User Store durch einen Standarddienst wie Lightweight Directory Access Protocol (LDAP) oder relationale Datenbank implementiert.

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle zum User Store.

Qualitäts-/Leistungsmerkmale

- Hohe Sicherheit.
- Hohe Verfügbarkeit

Offene Punkte/Probleme/Risiken

Diese Komponente könnte auch als eigene oder Teil einer eigenen Komponente realisiert werden, die für die personalisierten Funktionen des Portals zuständig ist.

2.8.4 Policy Store

Beschreibung

Der Policy Store ermöglicht das Speichern der Zugriffsregeln für das umwelt.info Portal.

Im Policy Store werden die Regeln abgelegt, die bei der Autorisierung von Anfragen an das umwelt.info Portal erforderlich sind. Eine solche Regel könnte etwa beinhalten, dass der Aufruf des „Data Check-In“ nur möglich ist, wenn dem aufrufenden Nutzer die Rolle „Metadaten-Redakteur*in” zugewiesen wurde.

Weitere Details

Der Policy Store kann durch einen Standarddienst wie LDAP oder relationale Datenbank implementiert werden, ist manchmal aber auch rein dateibasiert.

Eine Möglichkeit, Zugriffsregeln deklarativ zu definieren, bietet die Sprache XACML.

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle zum Policy Store.

Qualitäts-/Leistungsmerkmale

- Hohe Sicherheit
- Hohe Verfügbarkeit

2.8.5 Personalization Service

Beschreibung

Ermöglicht die direkte Verwaltung der personalisierten Inhalte durch registrierte Nutzer*innen im umwelt.info Portal, wie etwa:

- Benachrichtigungen: die Benachrichtigungen werden nicht durch die registrierten Nutzer*innen des Portals gepflegt, sondern automatisch durch den Notification Generator erzeugt.
- Favoriten: die Favoriten werden durch die Nutzer*innen des Portals gepflegt.
- Gemerkte Suchen: Die gemerkten Suchen werden durch die Nutzer*innen des Portals gepflegt.

Die Nutzung dieses Dienstes durch Nutzer*innen bedingt die vorherige Anmeldung am umwelt.info Portal. Nutzer*innen können nur auf ihre eigenen Inhalte zugreifen (Ausnahme: die Favoritenlisten werden mit anderen Nutzer*innen geteilt).

Im Personalization Store werden die personalisierten Inhalte gespeichert.

Weitere Details

Die maximale Anzahl der gespeicherten Objekte per Nutzer / Inhaltskategorie sollte konfigurierbar sein.

Die gespeicherten personalisierten Inhalte müssen sich über einen Schlüssel den Profilen im User Store zuordnen lassen können (z. B. über den eindeutigen Nutzernamen).

Es sollte möglich sein, die personalisierten Inhalte nicht mehr vorhandener Nutzer*innen zu identifizieren oder automatisiert zu löschen.

Umgekehrt sollte es Nutzer*innen gemeldet werden, wenn beispielsweise vermerkte Metadaten aus den Favoriten nicht mehr existieren.

Schnittstelle(n)

Bereitgestellt:

- CRUD-Schnittstelle für Benachrichtigungen.
- CRUD-Schnittstelle für Favoriten.
- CRUD-Schnittstelle für gemerkte Suchen.

Benötigt:

- Schnittstelle zum Personalization Store.

Qualitäts-/Leistungsmerkmale

Eine JSON-basierte REST-Schnittstelle zur einfachen Einbindung in Browser-basierte Clients wäre vorteilhaft.

Offene Punkte/Probleme/Risiken

Diese Komponente könnte auch als eigene oder Teil einer eigenen Komponente realisiert werden, die für die personalisierten Funktionen des Portals zuständig ist.

2.8.6 Notification Generator

Beschreibung

Dieser Dienst generiert Benachrichtigungen für registrierte Nutzer*innen im umwelt.info Portal. Die Benachrichtigungen beziehen sich auf Statusänderungen der Favoriten und von gemerkten Suchen.

Der Notification Generator ist ein interner Dienst. Er reagiert auf ein bestimmtes Ereignis (z. B. die Zeit (wie jede Nacht um 0:00 Uhr, jeden Sonntag um 03:00 Uhr) oder auf Änderungen im Metadatenbestand). Beim Eintreffen des Ereignisses wird für die Favoriten der registrierten Nutzer*innen bestimmt, ob der betreffende Metadatensatz aktualisiert wurde. Wenn ja, wird eine Benachrichtigung erzeugt und an den Personalization Service zur Speicherung im Personalization Store gesendet. Für die gemerkten Suchen wird bestimmt, ob für eine Suche bei diesem Ereignis mehr Treffer gefunden werden als bei der letzten Prüfung einer gemerkten Suche. Wenn ja, wird eine Benachrichtigung erzeugt und an den Personalization Service zur Speicherung im Personalization Store gesendet

Weitere Details

Der Notification Generator sollte sich einfach um andere Benachrichtigungsarten erweitern lassen.

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle zum Triggern von Ereignissen

Benötigt:

- Schnittstelle zum Personalization Service

Qualitäts-/Leistungsmerkmale

- Fehlertoleranz

Offene Punkte/Probleme/Risiken

Diese Komponente könnte auch als eigene oder Teil einer eigenen Komponente realisiert werden, die für die Personalisierten Funktionen des Portals zuständig ist.

2.8.7 Personalization Store

Beschreibung

Ermöglicht das Persistieren der personalisierten Inhalte von registrierten Nutzer*innen im umwelt.info Portal: Benachrichtigungen, Favoriten, Gemerkte Suchen.

Der Zugriff auf die personalisierten Inhalte erfolgt über den Personalization Service.

Weitere Details

Die physische Speicherung der personalisierten Informationen kann in einer gemeinsamen Datenbank mit den Nutzerprofilen erfolgen oder in einer getrennten Datenbank, falls erforderlich.

Die gespeicherten personalisierten Inhalte müssen sich über einen Schlüssel den Profilen im User Store zuordnen lassen können (z. B. über den eindeutigen Nutzernamen).

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle zum Personalization Store.

Offene Punkte/Probleme/Risiken

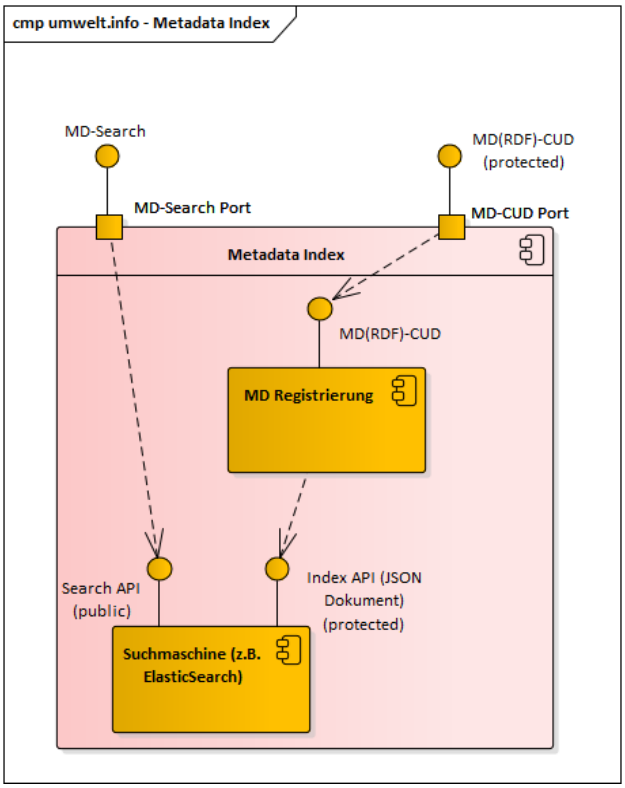
Diese Komponente könnte auch als eigene oder Teil einer eigenen Komponente realisiert werden, die für die personalisierten Funktionen des Portals zuständig ist.

2.9 Ebene 2 umwelt.info – Metadaten-Index (White Box)

Die Speicherung, Indizierung sowie Suche nach und Abfrage von Metadaten ist die wesentliche Aufgabe des Metadaten-Index. Das bedeutet, dass die Suchmaschine auch den primären Persistenzmechanismus für die Metadatensätze darstellt.

Vor allem eine einfache Volltext-Suche sowie eine Suche auf der Basis dezidierter Metadaten Eigenschaften (etwa unter Verwendung von Filtern/Facetten) sind wesentliche Funktionen des klassischen Metadaten-Index. Diese Funktionen können von einer Suchmaschine übernommen werden (z. B. Elasticsearch [14] oder Solr [15]). Die Metadaten-Dokumente werden im vorliegenden “Ressource Description Framework” (RDF) Format dem Metadaten-Index übergeben und dort volltextindiziert bzw. einzelne Elemente werden daraus extrahiert und separat indiziert (vgl. 5.1.1, Metadatenmodell).

Abbildung 10: Ebene 2 umwelt.info - Metadaten-Index (White Box)



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außer solchen, die bereits zuvor definiert worden sind).

2.9.1 Metadaten Registrierung

Beschreibung

Die Metadaten (MD) Registrierung bekommt die Metadaten im RDF-Format übergeben und transformiert diese in eine JSON Repräsentation, die für die Indizierung geeignet ist.

Weitere Details

Dabei werden die Werte der zu indizierenden Eigenschaften extrahiert und in eine angemessene JSON-Repräsentation für die Suchmaschine transformiert.

Weiter werden etwa Werte, die nicht als solche gespeichert sind, sondern als „linked resources“ (z. B. <dcterms:format rdf:resoure="http://publications.europa.eu/resource/authority/file-type/CSV“/>), de-referenziert und entsprechende Werte abgeleitet (im Bsp. „CSV“) und im JSON Dokument für die Suchmaschine hinterlegt.

Schnittstelle(n)

Bereitgestellt:

- MD(RDF)-Create/Update/Delete (CUD): Schnittstelle, die es erlaubt, Metadaten-Dokumente im RDF-Format erstmalig zu indizieren (create (C)), neu zu indizieren (update (U)) und zu löschen (delete (D)). Dabei wird aus der RDF-Repräsentation eine JSON Repräsentation erzeugt, die der

eigentlichen Suchmaschine zur Indizierung übergeben wird. Diese Schnittstelle ist „protected", d. h. nicht von außen (etwa aus dem Internet) zugänglich.

Benötigt:

- Index API: Native Schnittstelle der Suchmaschine zur Indizierung eines Metadaten-Dokumentes in JSON-Format. Der CUD-Teil dieser Schnittstelle ist ebenfalls „protected", d. h. nicht von außen (etwa aus dem Internet) zugänglich. Es können also keine Daten aus dem Internet verändert, wohl aber Daten gelesen werden.

Offene Punkte/Probleme/Risiken

Dereferenzierung von linked resources ist möglicherweise abhängig von externen Web-Diensten

2.9.2 Suchmaschine

Beschreibung

Bei der Suchmaschine handelt es sich um eine Software zur Suche nach den Metadaten, welche (wie zuvor beschrieben) gespeichert und anhand unterschiedlicher Eigenschaften indiziert worden sind.

Mittels Suchanfragen auf der Basis von Eigenschafts-Werten, liefert die Suchmaschine die gefundenen Metadaten, inklusive Titel, einem kurzen Auszug der Beschreibung sowie eine Liste von Verweisen auf möglicherweise relevante Dokumente.

Die Güte der Suchtreffer ist von besonderer Bedeutung. Daher sollte der fortlaufenden Optimierung der Treffer besondere Aufmerksamkeit zukommen.

Weitere Details

Ein wichtiger Aspekt ist die Sortierung der Ergebnisliste (etwa anhand des Datums der letzten Aktualisierung oder nach der Anzahl der exakten Treffer in einem Textfeld). Die Sortierung nach Relevanz hat für die Nutzenden eine besondere Bedeutung, da die relevantesten Treffer das Informationsbedürfnis der Nutzenden am schnellsten adressieren können.

Für die Sortierung nach Relevanz kommt in der Regel der Algorithmus BM25 [11] [12] zum Einsatz, der eine Gewichtung der Begriffe (anhand der Dokumentlänge und Vorkommen des Begriffs) für die Bewertung der Treffer bewirkt. Dieser Standardalgorithmus ist nicht festgeschrieben und sollte bei der Entwicklung der Suche und im Betrieb ständig auf seine Güte geprüft werden.

Mittels weiterer Techniken können Gewichtungen für bestimmte Felder vorgenommen werden oder bestimmte Datensätze besonders gewichtet werden (vgl. „Boosting" in 2.2.2). Diese Tätigkeit der Optimierung des Rankings wird „Relevance Engineering" genannt. Ein Praxisleitfaden ist zum Beispiel „Relevant Search" [16].

Dokumente werden üblicherweise im JSON - oder XML-Format gespeichert und abgefragt.

Schnittstelle(n)

Bereitgestellt:

- Index API: s.o.
- Search API: Native Schnittstelle der Suchmaschine zur Suche nach indizierten Dokumenten. Sie stellt verschiedene Möglichkeiten der Volltextsuche als auch z. B. einer Facetten-basierten Suche bereit.

- Bei Bedarf, Service der Begriffe liefert, die für eine Sucherweiterung verwendet werden können und von der Suchmaschine abgeleitet werden.

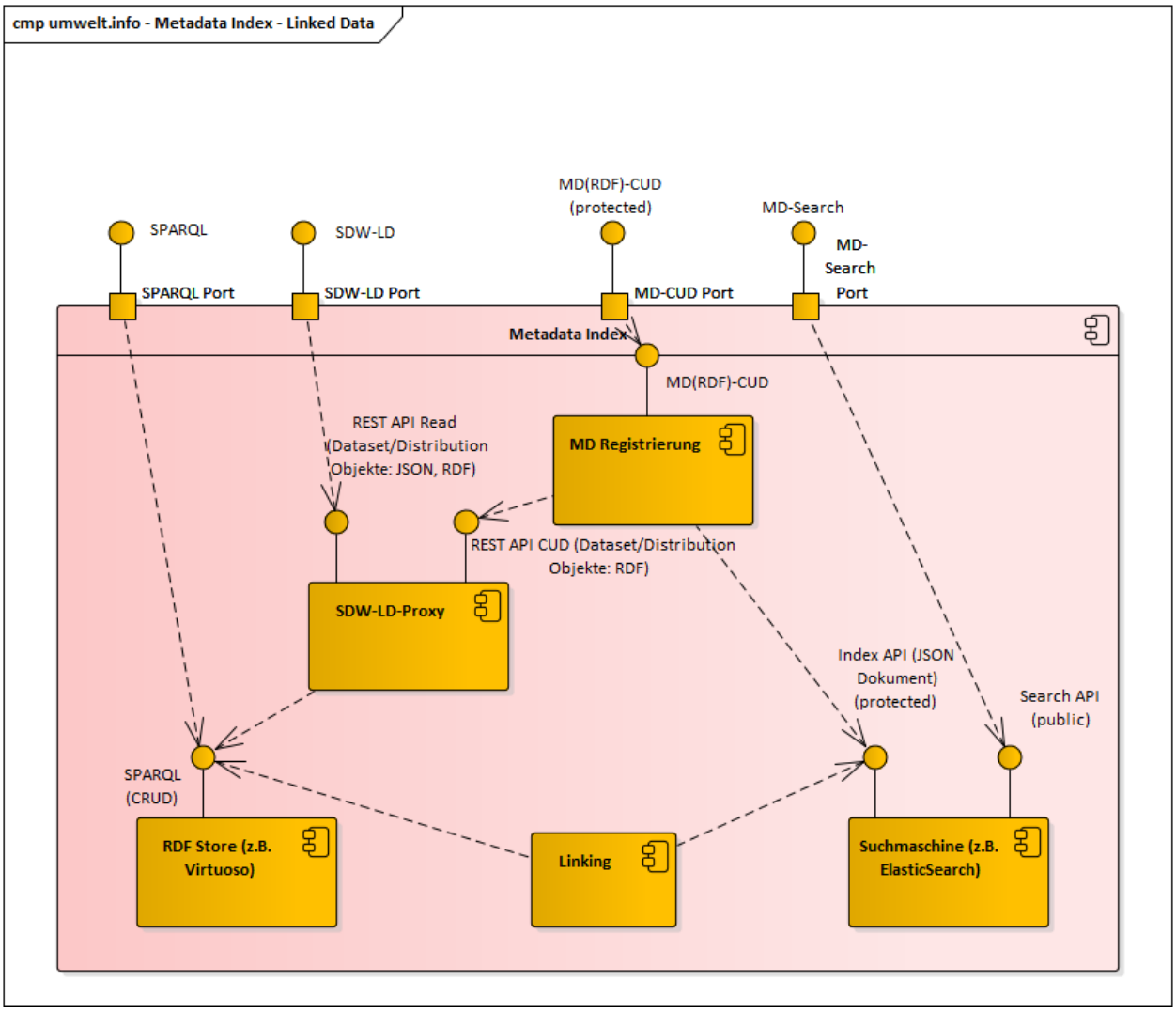
2.10 Ebene 2 umwelt.info – Metadata-Index – Option: Linked Data und Spatial Data on the Web (White Box)

Bei der Option Metadaten-Index (mit Linked Data und (Spatial) Data on the Web Unterstützung) werden zusätzlich zur vorhergehenden Lösung des Metadaten-Index die Metadaten in einem Triple-Store gespeichert, um Metadaten etwa über eine persistente HTTP URI identifizierbar zu machen und um Anfragen mittels (Geo)SPARQL durchführen zu können. Alle Zugriffe werden in einem der wichtigsten RDF-Serialisierungen (z. B. JSON-LD oder RDF/XML) zurückgegeben werden.

Für weitere Anwendungsmöglichkeiten (etwa für den LD-Service) sollte die Indizierung pro Metadatensatz in der Suchmaschine erweitert werden, mindestens um eine Indizierung der Anzahl ausgehender und eingehender Links, die das Ranking bei einer Suche beeinflussen soll.

Voraussetzung für viele Linked Data Anwendungen ist auch, dass zusätzlich zu den bestehenden Links weitere Links zwischen den Metadaten im TripleStore aufgebaut werden. Dieses kann von einem speziellen Prozess (Linking) übernommen werden, der auf dem Triple Store agiert. Dieser kann bei Bedarf asynchron im Hintergrund laufen, um den Harvesting Prozess nicht zu verlangsamen.

Abbildung 11: Ebene 2 Metadata-Index - Option: Linked Data und Spatial Data on the Web (White Box)



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außen denen die schon zuvor definiert und nicht modifiziert worden sind).

2.10.1 MD-Registrierung

Beschreibung

Erweiterung zur MD-Registrierung im Kapitel 2.9.1.

Weitere Details

Zusätzlich zur MD-Registrierung aus dem Ansatz ohne SDW / Linked Data übergibt diese Komponente das Metadaten-Dokument im RDF Format an die SDW-LD Proxy Komponente.

Schnittstelle(n)

Bereitgestellt:

- MD(RDF)-CUD: Schnittstelle, die es erlaubt, Metadaten-Dokumente im RDF-Format erstmalig zu indizieren (create (C)), neu zu indizieren (update (U)) und zu löschen (delete (D)). Dabei wird

aus der RDF-Repräsentation zusätzlich eine JSON Repräsentation erzeugt, die der eigentlichen Suchmaschine zur Indizierung übergeben wird. Diese Schnittstelle ist „protected", d. h. nicht von außen (etwa aus dem Internet) zugänglich.

Benötigt:

- Index API: Native Schnittstelle der Suchmaschine zur Indizierung eines Metadaten-Dokumentes in JSON-Format. Diese Schnittstelle ist ebenfalls „protected", d. h. nicht von außen (etwa aus dem Internet) zugänglich.
- SDW-LD REST API (CUD): Über diese REST-Schnittstelle werden die Metadaten konform zu den Anforderungen von (Spatial) Data on the Web (SDW) und Linked Data (LD) gespeichert und über eine persistente HTTP URI identifizierbar und im HTML-Format (sowie JSON(-LD)) zugreifbar gemacht. Das HTML ist für ein besseres Suchergebnis auf der Basis der schema.org und DCAT-AP Vokabulare annotiert. Die Ressourcen werden mit anderen Ressourcen verlinkt.

2.10.2 SDW-LD Proxy

Beschreibung

Diese Komponente stellt eine Spatial Data on the Web bzw. Linked Data Implementierung auf dem RDF (Triple) Store dar.

Weitere Details

Konkret stellt diese Komponente eine REST-Schnittstelle bereit, die die Metadaten in den wichtigsten Serialisierungen, neben RDF XML/JSON-LD, JSON und HTML (mit schema.org Annotationen), abgeben kann.

Die Ressourcen, die diese Komponente bereitstellt, bilden die wesentlichen DCAT-AP Entitäten ab, primär also „Dataset" und „Distribution".

Eine wesentliche Aufgabe der Komponente ist es, die RDF-Daten vorzuverarbeiten und zu harmonisieren. Hierzu gehört die Anwendung von konsistenten und bedeutungsvollen URI-Schemata, die Erzeugung eindeutiger URI´s und die Zuordnung (mapping) zu existierenden Objekten (resolution and linking).

Die API sollte es auch erlauben, nach verschiedenen Eigenschaften einfach zu filtern, wie etwa nach Begriffen, dem Dokumenttyp, dem Raumbezug, dem Aktualitätsdatum, etc. Sie sollte die Möglichkeit bieten, durch den Datenbestand entlang verschiedener hierarchisch organisierter Eigenschaftswerte zu navigieren (etwa Daten zu: Bundesland->Kreis->Stadt).

Schnittstelle(n)

Bereitgestellt:

- SDW-LD REST API (CUD): s. 2.10.1
- SDW-LD REST API (READ): Über diese Schnittstelle werden die Metadaten konform zu Anforderungen von Spatial Data on the Web (SDW) und Linked Data (LD) zugreifbar gemacht: Alle Ressourcen lassen sich über eine persistente HTTP URI identifizieren. Interaktionen finden auf der Basis des HTTP-Protokolls (REST) statt. Alle Ressourcen sind auffindbar über Internet-Suchmaschinen. Hierfür werden sie im HTML-Format zugreifbar gemacht, dass für ein besseres Suchergebnis, auf der Basis der schema.org bzw. DCAT-AP Vokabulare, annotiert ist. Die

Ressourcen sind mit anderen Ressourcen verlinkt bzw. lassen sich von anderen Ressourcen linken.

Benötigt:

- SPARQL (CRUD): Abfragen, Schreiben und Ändern von OAW-Tripeln in einem Triple-Store.

2.10.3 RDF (Triple) Store

Beschreibung

Ein sogenannter Triple Store fungiert als zentrale Datenbank. Hier werden die RDF Triple gespeichert. Der Zugriff erfolgt über SPARQL.

Weitere Details

Der SPARQL Zugangspunkt ist idealerweise auch nach außen offen. Der Triple Store erlaubt zusammen mit dem DCAT-AP/RDF basierten Metadatenmodell eine maximale Flexibilität für potenzielle Anwendungen, die darauf aufsetzen. Um Angriffe (z. B. „Denial-of -Service-Angriffe“, einfach durch Konstruktion komplexer Queries) zu verhindern, muss der Zugang allerdings beschränkt werden (mindestens einer Absicherung i.S.v. Ressourcenbeschränkung) oder gar doch nur über einen entsprechend erweiterten SDW-LD Service (s. 2.10.2) erlaubt werden. Die Funktionen zum Ändern der Daten (CUD) müssen ohnehin geschützt werden.

Schnittstelle(n)

Bereitgestellt:

- SPARQL: „SPARQL ist eine Graphen-basierte Abfragesprache für Abfragen von Inhalten aus dem Beschreibungssystem RDF (Resource Description Framework), das in Datenbanken zur Formulierung logischer Aussagen über beliebige Dinge genutzt wird. Der Name ist ein rekursives Akronym für SPARQL Protocol And RDF Query Language“ (Zitat aus: https://de.wikipedia.org/wiki/SPARQL).
  - SPARQL (CUD): der Teil von SPARQL, der das Schreiben und Ändern von OAW-Tripeln in einem Triple-Store ermöglicht.
  - SPARQL (READ): der Teil von SPARQL, der das Abfragen von Inhalten aus dem Beschreibungssystem RDF ermöglicht.

2.10.4 Linking

Beschreibung

Voraussetzung für viele Linked Data Anwendungen ist, dass zusätzlich zu den bestehenden Links weitere Links zwischen den Metadaten im Triple Store bzw. zu Thesaurus-Einträgen aufgebaut werden. Dieses kann von einem speziellen Prozess übernommen werden, der auf dem Triple Store agiert.

Weitere Details

Es gibt bereits eine Reihe von maschinellen Ansätzen zum automatisierten Verlinken von RDF-Daten. Diese basieren auf der Ermittlung der semantischen Nähe zweier Datensätze etwa auf der Basis gemeinsam verwendeter Keywords (etwa der aus einem gemeinsam verwendeten Thesaurus), den Zeitpunkt oder Zeitraum oder den geographischen Bezug. Hierzu gibt es auch

bereits verschiedene Tools (s. z. B. SILK (http://silkframework.org/)) oder auch LIMES (https://aksw.org/Projects/LIMES.html) für das Linken über den geographischen Bezug.

Da die Ressourcen in der vorliegenden Domäne auf DCAT-AP beruhen, vielfach einen Raum- und Zeitbezug besitzen und häufig Begriffe aus gleichen Thesauri verwenden, besteht eine gute Chance automatisch Links zwischen ihnen erzeugen zu können.

Es besteht auch die Möglichkeit, eine Komponente zu verwenden, die ein trainiertes NN repräsentiert, dass auf der Basis von Trainingsdatensätzen gelernt hat, Kurzbeschreibungen in den Metadaten bestimmten Konzepten einer Taxonomie zuzuweisen. Letzteres wird bereits im UBA für Forschungsberichte im Zusammenhang mit der UFORDAT praktiziert.

Die erzeugten Links werden im Triple-Store abgelegt und die Anzahl eingehender bzw. ausgehender Links wird auch im Index der Suchmaschine aktualisiert.

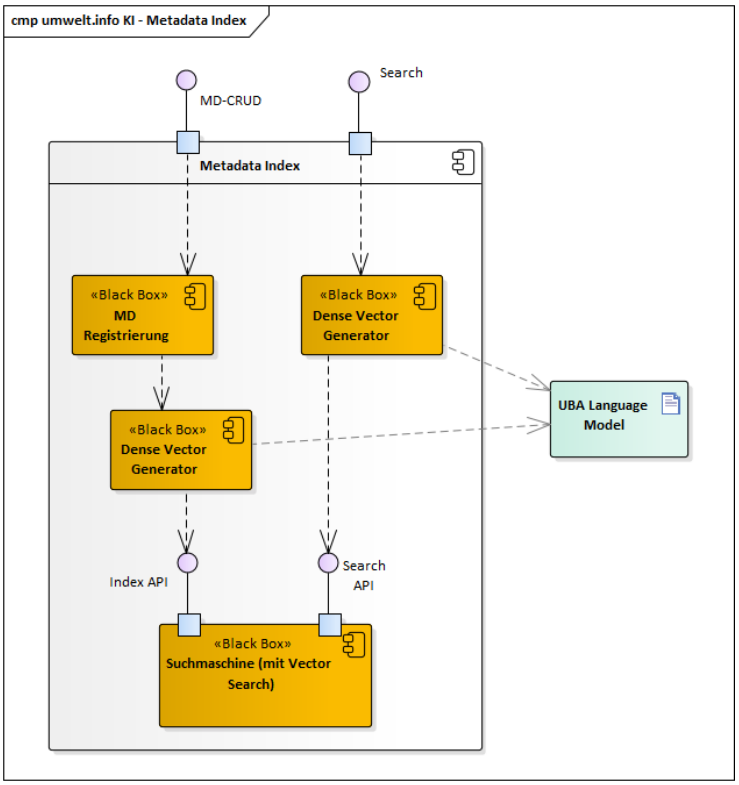
Schnittstelle(n)

Benötigt:

- SPARQL Schnittstelle des Triple-Stores (inklusive der Schreib-Operationen)

2.11 Ebene 2 umwelt.info – Metadaten-Index – Option: Sprachmodell (White Box)

Abbildung 12: Ebene 2 umwelt.info - Metadaten-Index - Option: Sprachmodell (White Box)



2.11.1 Dense Vector Generator

Beschreibung

Unter Verwendung eines Sprachmodells lassen sich Textfragmente als Vektoren repräsentieren, die den semantischen Kontext der vorkommenden Wörter miterfassen. Diese Vektorrepräsentation wird genutzt, indem die Vektoren für eine spezielle „Vector Search“ im Index mitgespeichert werden.

Weitere Details

Um diese Vektorrepräsentation nutzen zu können, müssen bei der Indexierung die zu indexierenden Vektoren für die zu indexierenden Textfragmente erzeugt werden und bei der Suche ein Vektor für den Suchtext.

Bei der Verwendung einer Vektorrepräsentation ist die Annahme, dass ein Sprachmodell die Vektoren so erzeugen kann, dass nützliche Kontextinformationen mit den im Text vorkommenden Wörtern erfasst werden (siehe 5.6.3).

Schnittstelle(n)

- Eingabe: Textfragment
- Ausgabe: Dense Vector
- Sprachmodell

2.11.2 Suchmaschine mit Vector Search

Beschreibung

Für die Option „Suche mit Sprachmodell“ ist darauf zu achten, dass der Index die Fähigkeit für einen „Ähnlichkeitsvergleich“ von Vektoren hat.

Weitere Details

Die benötigten Feldtypen und Funktionen werden von den Apache Lucene-basierten Suchmaschinen Elasticsearch und Sol derzeit umgesetzt und optimiert. Die Realisierung eines Ähnlichkeitsvergleiches ist alternativ, etwa für ältere Versionen auch unter Hinzunahme weiterer Softwareprodukte möglich (vgl. Haystack 2.2.10).

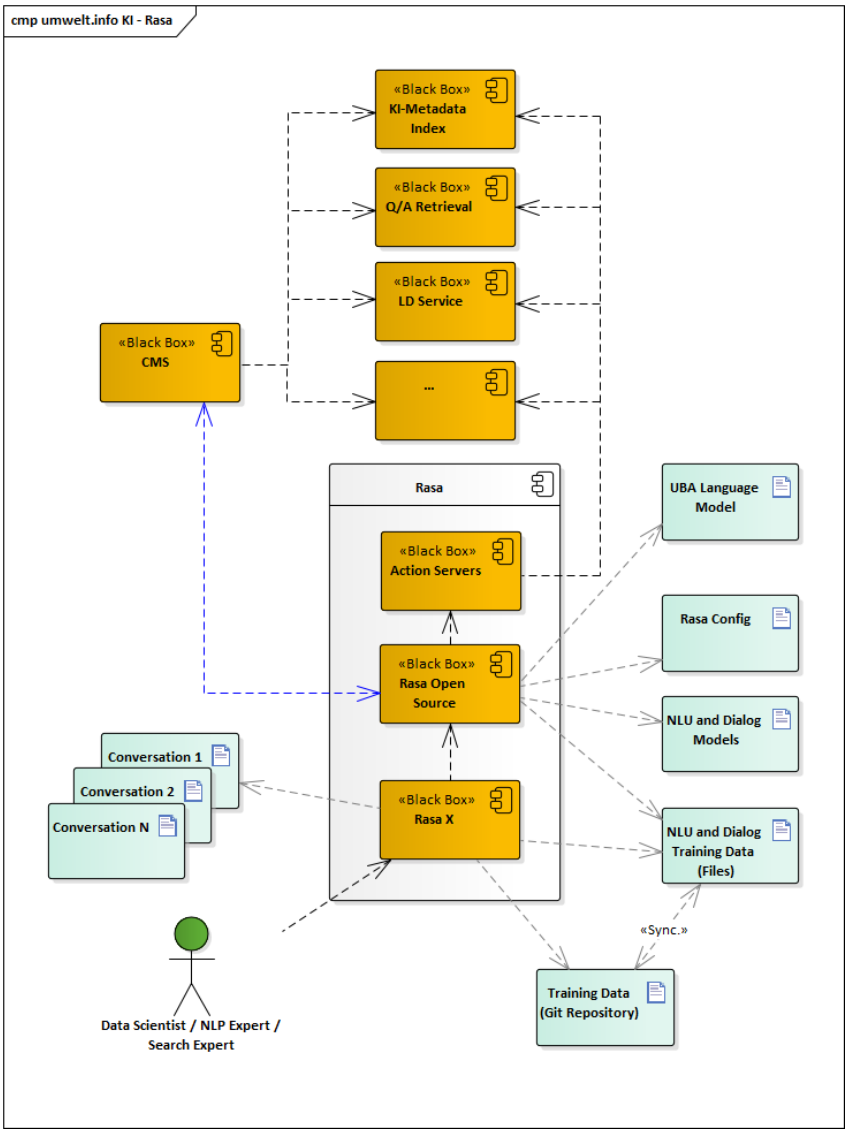
https://lucene.apache.org/core/9_0_0/changes/Changes.html

Implementierungen:

- Elasticsearch: „nearest-neighbor search across high-dimensionality vectors“ (https://lucene.apache.org/core/corenews.html#apache-lucenetm-920-available).
- Solr: „Dense Vector Neural" Search through DenseVectorField field Type and K-Nearest-Neighbor (KNN) Query Parser“ (https://solr.apache.org/news.html, Solr 9.0.0 Release Highlights)

2.12 Ebene 2 umwelt.info – Rasa (White Box)

Abbildung 13 Ebene 2 umwelt.info – Rasa (White Box)



Quelle: eigene Darstellung, con terra GmbH

2.12.1 Rasa open source

Beschreibung

Rasa open source ist die Kernkomponente von Rasa. Sie ermöglicht es einem Programm oder einer Website Texteingaben mittels NLP zu verarbeiten und Dialoge zu führen. Dazu erkennt Rasa Intents und Entities in Texteingaben (Natural Language Understanding), sowie die jeweils nächste Aktion im Dialogablauf (Dialogsteuerung). Die Kernkomponente von Rasa stellt verschiedene Tools für die beiden Vorgänge des Trainings und der Dialogführung bereit. Dazu benötigt Rasa open source die abgebildeten Artefakte: Konfigurationsdatei, Trainingsdaten für die Dialogsteuerung, sowie Trainingsdaten für das Natural Language Understanding.

Weitere Details

Während des Trainings werden Modelle trainiert, die für das Natural Language Understanding (Intent- und Entity Recognition) und die Dialogführung genutzt werden. Der Agent der Rasa

Hauptkomponente nutzt diese Modelle für die Steuerung des Dialogs. Dabei besteht ein Dialog aus Intents und Actions.

Es ist auch möglich, Dialoge auf Basis von Regeln zu definieren.

Es ist auch möglich, die Dialogsteuerung mittels Konfiguration zu deaktivieren. So kann Rasa auch ausschließlich für die Analyse von Texteingaben mittels NLU genutzt werden (etwa zur Klassifikation von Sucheingaben im Suchfeld).

Mit einer speziellen Konfiguration ist es möglich, ein eigenes Sprachmodell (etwa BERT-basiert) einzusetzen.

https://rasa.com/docs/rasa/nlu-training-data

https://rasa.com/docs/rasa/stories

https://rasa.com/docs/rasa/rules

Schnittstelle(n)

Bereitgestellt:

- Tools für: Betrieb, Training, Entwicklung und Test
- Schnittstellen für die Anbindung

Benötigt:

- Action Server für Custom Actions

2.12.2 Action Server

Beschreibung

Wenn das Dialogsystem der Rasa Hauptkomponente (Rasa Open Source) eine bestimmte Action für die Fortsetzung des Dialogs erkannt hat, sendet es für die Verarbeitung spezieller Actions eine Anfrage an den Action Server. Dieser kann Actions zuordnen und Custom Actions (eigene Implementierungen) ausführen. So kann das Dialogsystem nicht nur Nachrichten als Antworten senden, sondern auch beliebigen Code als Reaktion auf eine eingehende Nachricht ausführen.

Weitere Details

Eine Anfrage an den Action Server enthält den Namen der Action, um die jeweilige Action einer Custom Action-Implementierung zuzuordnen.

Custom Actions können unabhängig vom Dialog beliebigen Code ausführen, verschiedene APIs ansprechen und den weiteren Dialog beeinflussen.

Mit der Antwort des Action Servers setzt der Agent nach Verarbeitung einer Custom Action den Dialog fort. Dabei sind Dialoge in Rasa eine Folge von Events und folglich ist auch der Rückgabewert einer Custom Action ein Event.

In umwelt.info müssen verschiedene Funktionalitäten als Custom Actions implementiert werden, wenn diese von der Dialogsteuerung ausgehend getriggert werden sollen. Dabei sollten die jeweiligen Actions nur die jeweilige API ansprechen (vgl. Abbildung 12), da die eigentlichen Funktionen auch unabhängig von Rasa nutzbar sein sollen. Zum Beispiel sollte Rasa für eine Empfehlung eine RS-Action implementieren, dabei aber nur die API der umwelt.info-RS-Komponente nutzen.

Für die Implementierung von Custom Actions als aufrufende Komponenten anderer Komponenten in umwelt.info eignet sich der Rasa Action Server, für den mittels des „rasa sdk“ einfache Actions implementiert werden können.

https://rasa.com/docs/action-server/

https://rasa.com/docs/action-server/actions

https://rasa.com/docs/action-server/pages/action-server-api/

https://rasa.com/docs/action-server/sdk-actions

Schnittstelle(n)

Bereitgestellt:

- Custom Actions zur Ausführung bel. Aktionen als Reaktion auf eine eingehende Nachricht (via „rasa sdk“)

Benötigt:

- Rasa Open Source

2.12.3 Rasa X

Beschreibung

Rasa X ist ein Werkzeug zur Wartung und Entwicklung von Conversational AI-Interfaces, im Sinne des Conversation-Driven Development (CCD).

Es enthält eine Benutzeroberfläche für Annotation von getrackten Daten, zur Erstellung von Trainingsdaten, sowie für das Training und die Verwaltung von Modellen.

Weitere Details

CCD ist ein Paradigma für den Entwurf und das Design eines natürlichsprachlichen Interfaces. Demnach sollte so früh wie möglich bei der Gestaltung von Dialogen auf reale Nutzerdaten zurückgegriffen werden, da nicht vorhersehbar ist, wie diese Dialoge aussehen sollten. Rasa empfiehlt daher den CCD-Ansatz und bietet mit Rasa X eine Komponente zur Unterstützung der Dialogentwicklung. Erleichtert wird damit die Sammlung und Annotation von Trainingsdaten, sowie das Verwalten, Trainieren und Einsetzen von neuen Modellen.

Für den Einsatz von Rasa X wird eine mit Enterprise-Lizenzierung benötigt. Der Support für die vormals verfügbare „Community Edition“ wurde eingestellt [17] (Stand 08.06.2022).

Schnittstelle(n)

Bereitgestellt:

- Benutzerschnittstelle für CCD

Benötigt:

- Rasa Open Source

2.13 Ebene 3 umwelt.info – Metadata Harvesting und Crawling – Harvester bzw. Crawler (White Box)

Die individuellen Metadata Harvester und Crawler übernehmen die eigentliche Metadaten-Ermittlung und das „semantische Anreichern" der Metadaten.

Nach [2] werden „Harvester" als Software-Bausteine verstanden, die Metadaten über existierende Software-Schnittstellen (APIs) mit bekannten Abfragemöglichkeiten und (Meta-) Datenprofilen ermitteln (s. Kapitel 3.2 (Kontextabgrenzung) im Umsetzungskonzept). Die Variabilität der Schnittstellen macht es notwendig, ein sehr gut konzipiertes Harvester-Konzept zu erstellen und verschiedene Harvester einzusetzen. Die Harvester sollten in der Lage sein, Filter anzuwenden, evtl. bereits Dubletten zu erkennen und zu behandeln, Schema- und Semantik-Tests durchzuführen und idealerweise die Datenqualität zu verbessern (vgl. [2]). Dies beinhaltet auch die Erzeugung einer Liste der Identifier aller Metadatensätze die später verwendet wird, um etwa Metadatensätze im System zu löschen, die in der Datenquelle nicht mehr vorhanden sind.

Nach [2] werden „Crawler" als Softwarebausteine verstanden, die ein Verzeichnis inklusive Unterverzeichnisse (bekannte Web- oder etwa UNC-Verzeichnisse) sukzessive durchsuchen und

- entweder die Metadaten aus Dateien mit bekanntem Metadatenmodell in das interne Metadatenmodell konvertieren - oder -
- die Metadaten aus Dateien mit bekanntem Datenmodell ableiten und in das interne Metadatenmodell konvertieren - oder -
- rudimentäre Metadaten aus Dateien mit unstrukturierten Daten ableiten

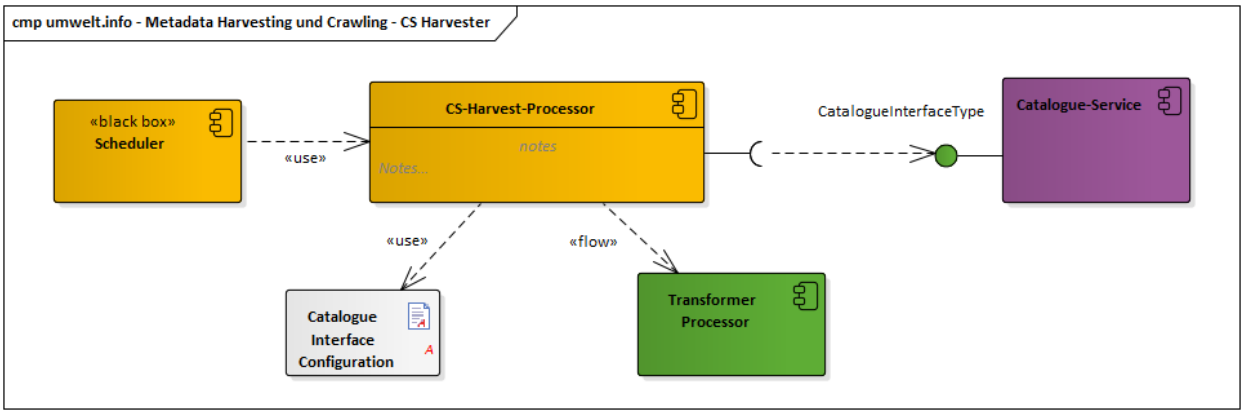
Anschließend werden die resultierenden Metadaten einem Transformer übergeben. Wichtig ist dabei, dass die Metadaten Informationen enthalten, um auf die (beschriebene) Daten-(Datei-) Ressource wieder zugreifen zu können (vgl. [2]).

2.13.1 CS-Harvester (Catalogue Services)

Hier handelt es sich um Harvester, die Metadaten über web-basierte Catalogue-Services mit bekannter Schnittstelle und bekanntem Metadatenprofil ermitteln.

Zu den „Catalogue Services" gehören die Dienste nach den INSPIRE Spezifikationen [5], nach der OGC Catalogue Services Spezifikation [4], der "Open Archives Initiative Protocol for Metadata Harvesting"-Spezifikation oder einem anderen (bekannten) Catalogue Interface (z. B. CKAN) zur Abfrage von Metadaten, dessen Web-Schnittstelle und Metadatenmodell bekannt ist. Die ermittelten Metadaten werden zur Transformation in das interne Metadatenschema an einen Transformer weitergeleitet.

Abbildung 14: Ebene 3 umwelt.info – Metadata Harvesting und Crawling – CS Harvester (White Box)



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außer denen die schon zuvor definiert worden sind).

2.13.1.1 CS-Harvest-Processor

Beschreibung

Der CS-Harvest-Processor führt das Harvesting durch, kennt die Web-Schnittstelle des Catalogue Service‘, parametrisiert und führt die Anfragen aus, die aus der Catalogue Interface Configuration gelesen wurden.

Weitere Details

Es werden die Metadaten abgerufen und es wird geprüft, ob sie einem (dem Standard entsprechenden) verpflichtenden Satz von Elementen entsprechen. Es können auch spezifische Qualitätsverbesserungen vorgenommen werden.

Damit ist der für ein Metadatenformat zuständige Transformer dann in der Lage, die Metadaten in das interne Format zu überführen.

Diese Komponente kann nicht sicherstellen, die genauen Lizenzbedingungen zu prüfen. Die Komponente kann lediglich die im Eingangsformat mitgelieferten Lizenz-relevanten Eigenschaften auf das interne Metadatenmodell „mappen”.

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle, die einen Harvesting Auftrag entgegennimmt.

Benötigt:

- Externer Metadaten-Dienst („Catalogue Service“) mit bekannter Web-Catalogue-Schnittstelle und bekanntem Metadatenmodell. Die Metadaten verweisen auf die eigentlichen Daten (vgl. [2]).
- Transformer-Dienst, an den die extrahierten Metadaten weitergeleitet werden.

Qualitäts-/Leistungsmerkmale

- Die Harvester bzw. Crawler sollten durch die Portalbetreiber (etwa mittels einer Konfigurationsdatei) konfigurierbar sein, so dass Anfragen an eine zu harvestende Datenquelle angepasst werden können.
- Die Schnittstellen sollten idealerweise die Möglichkeit besitzen, lediglich die Daten/Metadaten anzufordern, die sich seit dem letzten „Harvesting" geändert haben, um nicht immer alle Daten/Metadaten der Daten-Ressource wiederholt harvesten zu müssen.

2.13.1.2 Harvester Configuration

Beschreibung

Die Harvester Configuration enthält Konfigurationsinformationen für den Harvest-Processor.

Weitere Details

Hier werden Typ und Endpunkt (Zugriffpunkt) des externen Dienstes, Anfrageinformationen, aber auch „mapping" Informationen hinterlegt, die nötig sind, um von den externen Metadaten auf das interne Metadatenformat zu „mappen".

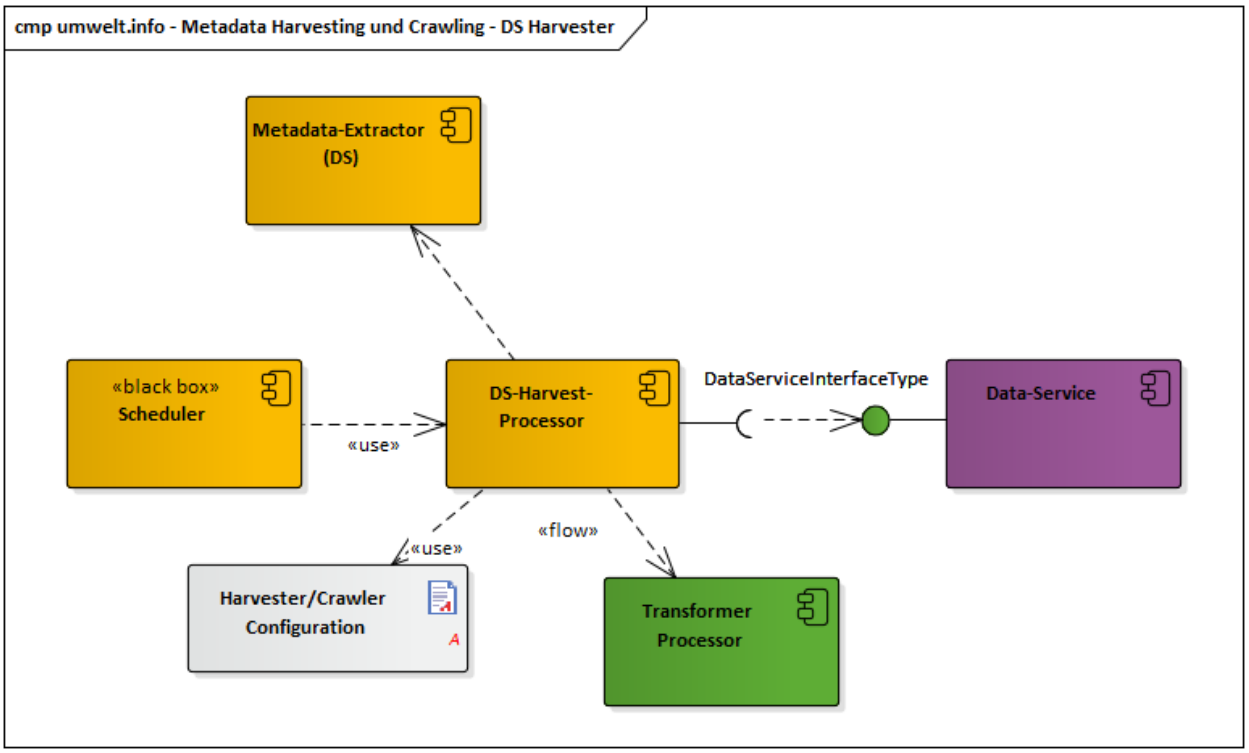
Offene Punkte/Probleme/Risiken

Benötigte Konfigurationsinformationen

2.13.2 DS-Harvester (Daten-Services)

Ein Daten-Service (DS)-Harvester dient der Ermittlung von Metadaten aus Daten-Diensten mit bekannter Schnittstelle und bekannter Service Beschreibung (vgl. [2]).

Abbildung 15: Ebene 3 umwelt.info - Metadata Harvesting und Crawling - DS-Harvester (White Box)



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außen denen die schon zuvor definiert worden sind).

2.13.2.1 DS-Harvest-Processor

Beschreibung (vgl. [2])

Führt das Harvesting durch, kennt die Web-Schnittstelle, führt die Anfragen aus, die er vom Harvest/Crawler-Manager mitgeteilt bekommt, überlässt die Extraktion der Metadaten dem Metadata-Extractor.

Weitere Details

Es werden primär die Service Metadaten bekannter Daten-Service Beschreibungen (z. B. ein OGC-Service „Capabilities" Dokument) abgerufen und durch den Metadata-Extractor interpretiert, der daraus die erforderlichen Metadaten ableitet (s. u.).

Zu den Daten-Diensten gehört etwa ein Service entsprechend den INSPIRE Technical Guidelines oder ein OGC Web Service (etwa ein Web Feature Service (OGC WFS) oder ein Web Coverage Service (OGC WCS)). Der Daten-Dienst erlaubt üblicherweise eine weitergehende Suche und einen Zugriff auf die einzelnen Datensätze der Quelle.

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle, die einen Harvesting Auftrag entgegennimmt.

Benötigt:

- Externer Daten-Dienst („Data Service“) mit bekannter Web-Service-Schnittstelle und bekannter Service Beschreibung.
- Metadata-Extractor Schnittstelle, die die externe Service-Beschreibung entgegennimmt und die Metadaten zurück liefert.
- Transformer-Dienst, an den die extrahierten Metadaten weitergeleitet werden.

Qualitäts-/Leistungsmerkmale

- Die Harvester bzw. Crawler sollten durch die Portalbetreiber konfigurierbar sein, so dass etwa die Anfrage an eine zu harvestende Datenquelle konfiguriert werden kann (vgl. [2]).
- Diese Komponente kann nicht sicherstellen, die genauen Lizenzbedingungen zu prüfen. Die Komponente kann lediglich die im Eingangsformat mitgelieferten Lizenz-relevanten Eigenschaften auf das interne Metadatenmodell „mappen”.

Offene Punkte/Probleme/Risiken

Auch bei bekannten Schnittstellen können Interoperabilitätsprobleme auftreten, die häufig spezifische Anpassungen an eine Datenquelle notwendig machen.

2.13.2.2 Metadata-Extractor (DS)

Beschreibung (vgl. [2])

Leitet aus der Service Beschreibung des Data-Service die Metadaten entsprechend dem internen Metadatenmodell ab. Die Metadaten verweisen auf den eigentlichen Daten-Dienst.

Weitere Details

Der Metadata-Extractor versucht die Metadaten aus einer Service-Beschreibung eines Daten-Dienstes abzuleiten, die semantisch das interne Metadatenmodell erfüllen, und in ein solches Format zu bringen auf dessen Basis ein entsprechender Transformer in der Lage ist, einen dem internen Metadaten-Format entsprechenden Metadatensatz zu erzeugen. Es werden primär die Beschreibungen der Daten aus der Service-Beschreibung extrahiert, auf denen der Daten-Dienst „operiert“. Im Wesentlichen geht es dabei darum, Collections, Layer, Feature Types, Coverage Types zu identifizieren, als Datensätze mit Metadaten zu beschreiben und die Zugriffsinformationen (die in den Metadaten zu hinterlegen sind) ebenfalls aus den Service-Metadaten zu extrahieren.

Schnittstelle(n)

Bereitgestellt:

Schnittstelle, die die externe Service-Beschreibung entgegennimmt und die Metadaten zurückliefert.

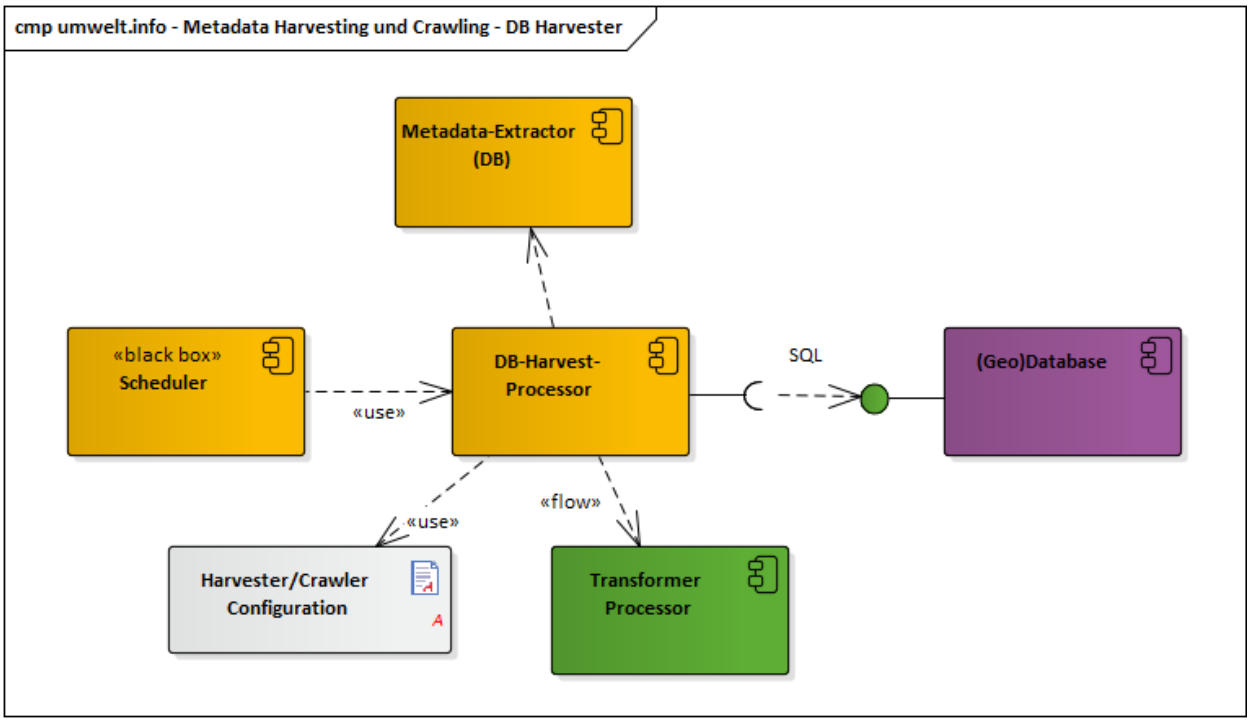
Offene Punkte/Probleme/Risiken

Die Metadaten zu den Collections, Layern, Feature Types, Coverage Types sind häufig sehr rudimentär gepflegt, so dass die Ableitung eines Metadatensatzes schwierig sein kann.

2.13.3 DB-Harvester (Datenbank-Harvester)

Ein Datenbank-Harvester dient der Ermittlung und Indizierung von (Geo-)Datenbanken (vgl. [2]).

Abbildung 16: Ebene 3 umwelt.info - Metadata Harvesting und Crawling - DB Harvester (White Box)



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außer denen die bereits zuvor definiert worden sind).

2.13.3.1 DB-Harvest-Processor

Beschreibung

Führt das Harvesting durch, kennt (im Idealfall) die Struktur der relationalen Datenbank, führt die SQL-Anfragen aus, die er vom Harvest/Crawler-Manager zugewiesen bekommt, übergibt die Antwort der SQL-Abfrage zur Extraktion der Metadaten an den Metadata-Extractor und übergibt anschließend die zurückgelieferten Metadaten via Transformer an den Metadaten-Index.

Weitere Details

Idealerweise sollten die zu harvestenden Einträge (z. B. Feature Types einer (Geo)Datenbank oder Einträge einer Forschungs- oder Stoffdatenbank) idealerweise als „View“ vom Datenbank-Betreiber bereitgestellt werden. Der DB-Harvest-Processor führt die eigentliche Abfrage auf die (Geo-)Datenbank durch. Er kennt die auszuführende SQL-Anfrage aus der Konfiguration. Er überlässt die Extraktion der Metadaten aus der Ergebnismenge dem Metadata-Extractor (DB).

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle, die einen Harvesting Auftrag entgegennimmt.

Benötigt:

- Externe (Geo-)Datenbank mit SQL-Schnittstelle (relationales Datenmodell).
- Metadata-Extractor Schnittstelle, die die externe Service-Beschreibung entgegennimmt und die Metadaten zurückliefert.
- Transformer-Dienst, an den die extrahierten Metadaten weitergeleitet werden.

Qualitäts-/Leistungsmerkmale

- Die Harvester sollten durch die Portalbetreiber konfigurierbar sein, so dass etwa die Anfrage an eine zu harvestende Datenquelle angepasst werden kann.
- Die Schnittstellen sollten idealerweise die Möglichkeit besitzen, lediglich die Daten anzufordern, die sich seit dem letzten „Harvesting" geändert haben, um nicht immer alle Einträge der Datenbank wiederholt harvesten zu müssen.

Offene Punkte/Probleme/Risiken

- Im Falle, dass keine View bereitgestellt wird, ist die Kenntnis der relationalen Struktur, deren Inhalte und die Erzeugung einer eventuell komplexen SQL-Anfrage notwendig.
- Der Zugriff auf die Datenbank muss vom Betreiber gewährt werden.

2.13.3.2 Metadata-Extractor (DB)

Beschreibung

Leitet aus den Ergebnissen der Abfrage auf die (Geo-)Datenbank die Metadaten entsprechend dem internen Metadatenmodell ab.

Weitere Details

Der Metadata-Extractor ermittelt aus den strukturierten Daten der Datenbank Anfrage (üblicherweise eine Tabelle) die Metadaten ab, die sich semantisch auf das interne Metadatenmodell (DCAT-AP.de) „mappen" lassen. Damit lässt sich dann ein dem Metadatenmodell entsprechender Metadatensatz erzeugen. Der Metadata-Extractor ermittelt die Metadaten auf der Basis eines konfigurierbaren Satzes an Regeln aus dem Ergebnis der SQL-Anfrage. Diesen Satz an Regeln liest der Harvester aus der ihm zugehörigen Konfiguration.

Diese Komponente kann nicht sicherstellen, die genauen Lizenzbedingungen zu prüfen. Die Komponente kann lediglich die mitgelieferten Lizenz-relevanten Eigenschaften auf das interne Metadatenmodell „mappen".

Schnittstelle(n)

Bereitgestellt:

Schnittstelle, die die externe Ergebnis-Tabelle (z. B. einen DB-Cursor) entgegennimmt und die Metadaten zurück liefert.

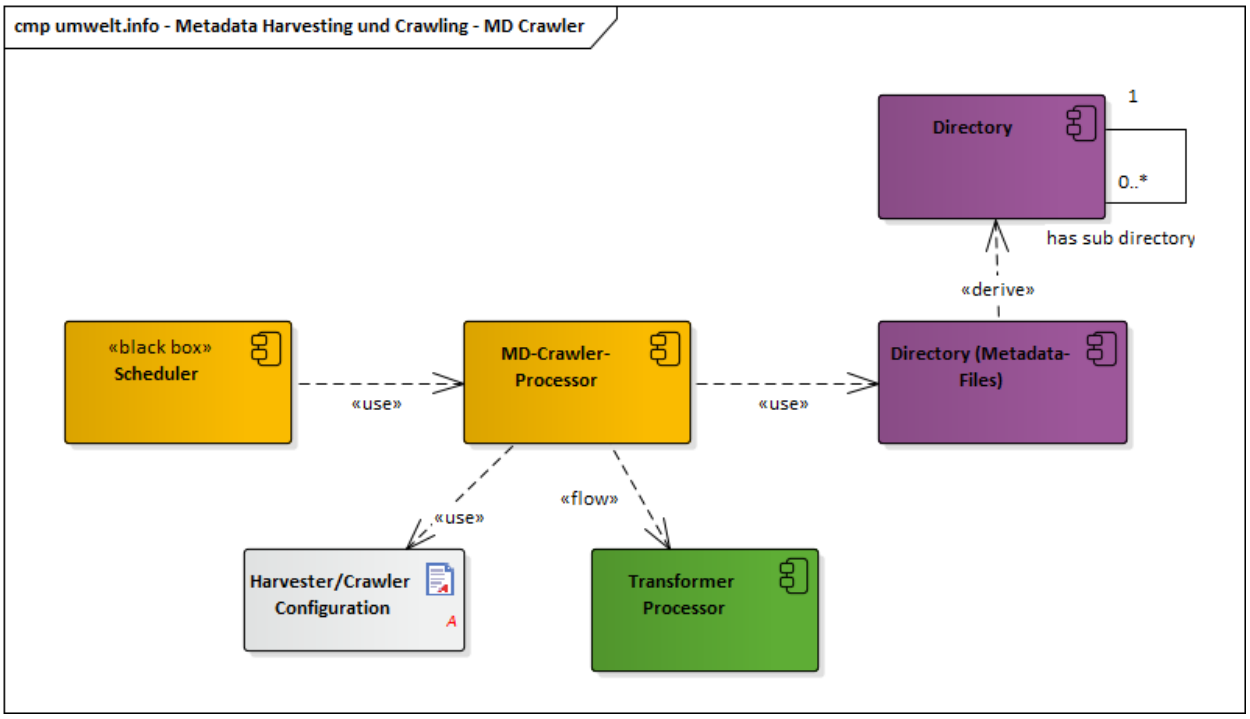
Offene Punkte/Probleme/Risiken

siehe DB-Harvest-Processor

2.13.4 MD-Crawler (Metadaten-Crawler)

Der MD-Crawler durchsucht Verzeichnisse mit Dateien, die Metadaten mit bekanntem Metadatenprofil enthalten. Ein Metadaten-Crawler dient der Ermittlung und Indizierung von Metadaten aus Dateien (in Web- oder etwa UNC-Verzeichnissen) mit bekannten Metadatenprofilen. Zu diesen Metadatenprofilen gehört etwa ISO19115, Dublin Core oder das Esri-Metadatenformat (vgl. [2]).

Abbildung 17: Ebene 3 umwelt.info - Metadata Harvesting und Crawling - MD-Crawler (White Box)



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außen denen die bereits zuvor definiert worden sind).

2.13.4.1 MD-Crawler-Processor

Beschreibung

Führt das Crawlen durch, weiß wie ein Web-Verzeichnis bzw. ein UNC-Verzeichnis (inklusive Unterverzeichnisse) sukzessive zu durchsuchen ist, dessen Zugriffsinformationen er aus der Harvester bzw. Crawler Configuration gelesen hat (vgl. [2]).

Der Crawler übergibt die Metadaten dann an einen Transformer.

Weitere Details

Es werden primär die Metadaten abgeholt und es wird geprüft, ob sie im Wesentlichen dem zu liefernden Metadatenformat entsprechen. Es können auch spezifische Qualitätsverbesserungen vorgenommen werden.

Damit ist der für ein Metadatenformat zuständige Transformer (s.o.) dann in der Lage, die Metadaten in das interne Format zu überführen.

Diese Komponente kann nicht sicherstellen, die genauen Lizenzbedingungen zu prüfen. Die Komponente kann lediglich die im Eingangsformat mitgelieferten Lizenz-relevanten Eigenschaften auf das interne Metadatenmodell „mappen".

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle, die einen Crawling Auftrag entgegennimmt.

Benötigt:

- Externes Web-Verzeichnis bzw. UNC-Verzeichnis (inklusive Unterverzeichnisse), das Dateien mit bekanntem Metadatenmodell enthält. Die Metadaten sollten auf die eigentlichen Daten verweisen (vgl. [2]).
- Transformer-Dienst, an den die extrahierten Metadaten weitergeleitet werden.

Qualitäts-/Leistungsmerkmale

- Die Crawler sollten durch die Portalbetreiber konfigurierbar sein, so dass etwa die Anfrage an eine zu crawlende Datenquelle konfiguriert werden kann.
- Beim Crawlen sollten idealerweise lediglich die Metadaten abgeholt werden, die sich seit dem letzten „Crawling" geändert haben, um nicht immer alle Metadaten wiederholt crawlen zu müssen.

2.13.4.2 Harvest/Crawler Configuration

Beschreibung

Die Harvest/Crawler Configuration enthält Konfigurationsinformationen für den Crawler-Processor.

Weitere Details

Hier werden Web-Verzeichnis bzw. UNC-Verzeichnis (Zugriffpunkt) festgelegt, aber auch „mapping" Informationen, um von den externen Metadaten oder Daten auf das interne Metadatenformat zu „mappen".

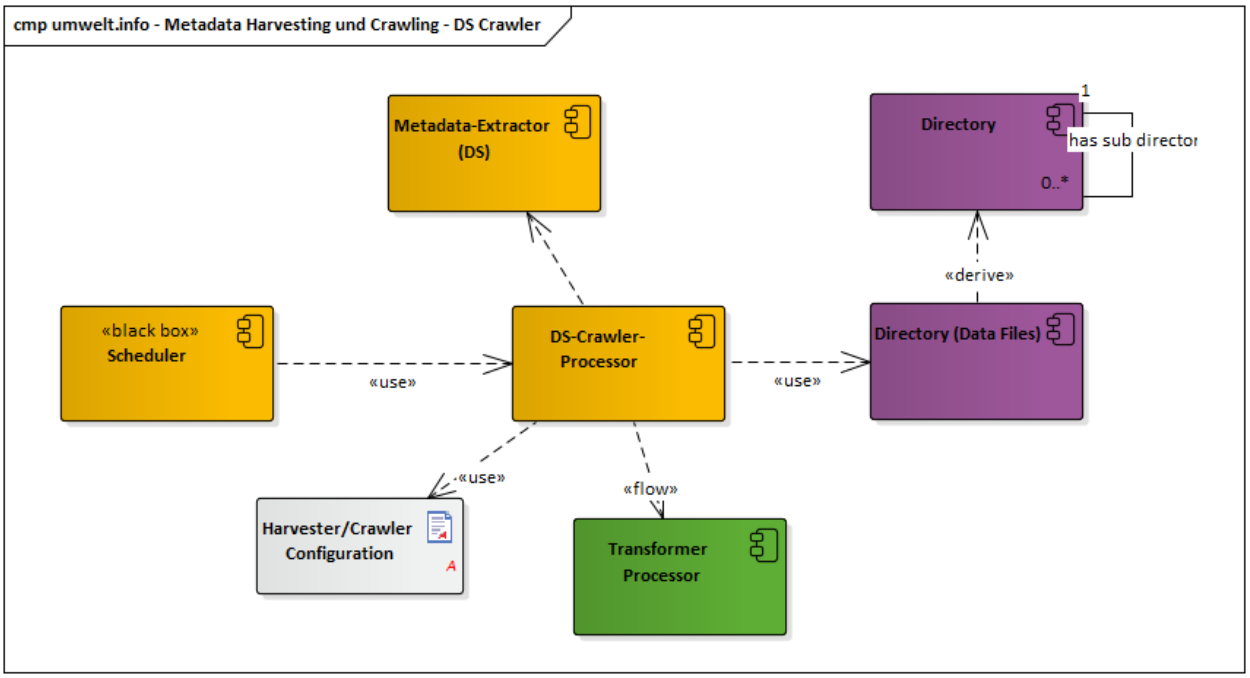
Offene Punkte/Probleme/Risiken

Benötigte Konfigurationsinformationen

2.13.5 DS-Crawler (Daten-Services)

DS-Crawler sind Crawler von Verzeichnissen mit Dateien, die Daten mit bekanntem Datenprofil enthalten (vgl. [2]).

Abbildung 18: Ebene 3 umwelt.info - Metadata Harvesting und Crawling - DS-Crawler



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außer denen die schon zuvor definiert worden sind).

2.13.5.1 DS-Crawler Processor

Beschreibung (vgl. [2])

Führt das Crawlen durch, weiß wie ein Web-Verzeichnis bzw. ein UNC-Verzeichnis (inklusive Unterverzeichnisse), welches er vom Harvest/Crawler-Manager mitgeteilt bekommt, sukzessive zu durchsuchen ist. Der Crawler überlässt die Ermittlung der Metadaten dem Metadata-Extractor (DS) und übergibt die Metadaten an den Transformer.

Weitere Details

Ein DS-Crawler dient der Ermittlung und Indizierung von Metadaten aus Dateien (in Web- oder etwa UNC-Verzeichnissen) mit bekannten strukturierten Daten (z. B. bekannte Datenprodukte, etwa bekannte Rasterdateien (z. B. GeoTIFF), Geländemodelle (z. B. SRTM-3), Shape-Files, ESA SAFE Files, Excel-Files, Statistiken mit bekanntem Format). Es wird versucht, nicht einzelne Daten zu beschreiben, sondern Gruppen von Daten zusammenzufassen. Hier sollten die Typen der Dateien recht leicht zu erkennen sein und sich automatisch Metadaten aus den bekannten Strukturen extrahieren lassen (inklusive eines Verweises auf die Quelle der Datei). Der Crawler ermittelt die benötigten Metadaten auf der Basis eines konfigurierbaren Satzes an Regeln aus den jeweiligen Daten Dateien.

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle, die einen Crawling Auftrag entgegennimmt.

Benötigt:

- Externes Web-Verzeichnis bzw. UNC-Verzeichnis (inklusive Unterverzeichnisse), das Dateien mit bekanntem Datenmodell enthält. Auf diese werden die extrahierten Metadaten verweisen (vgl. [2]).
- Metadata-Extractor Schnittstelle, die die Dateien mit bekannten Formaten entgegennimmt und die Metadaten zurückliefert.
- Transformer-Dienst, an den die extrahierten Metadaten weitergeleitet werden.

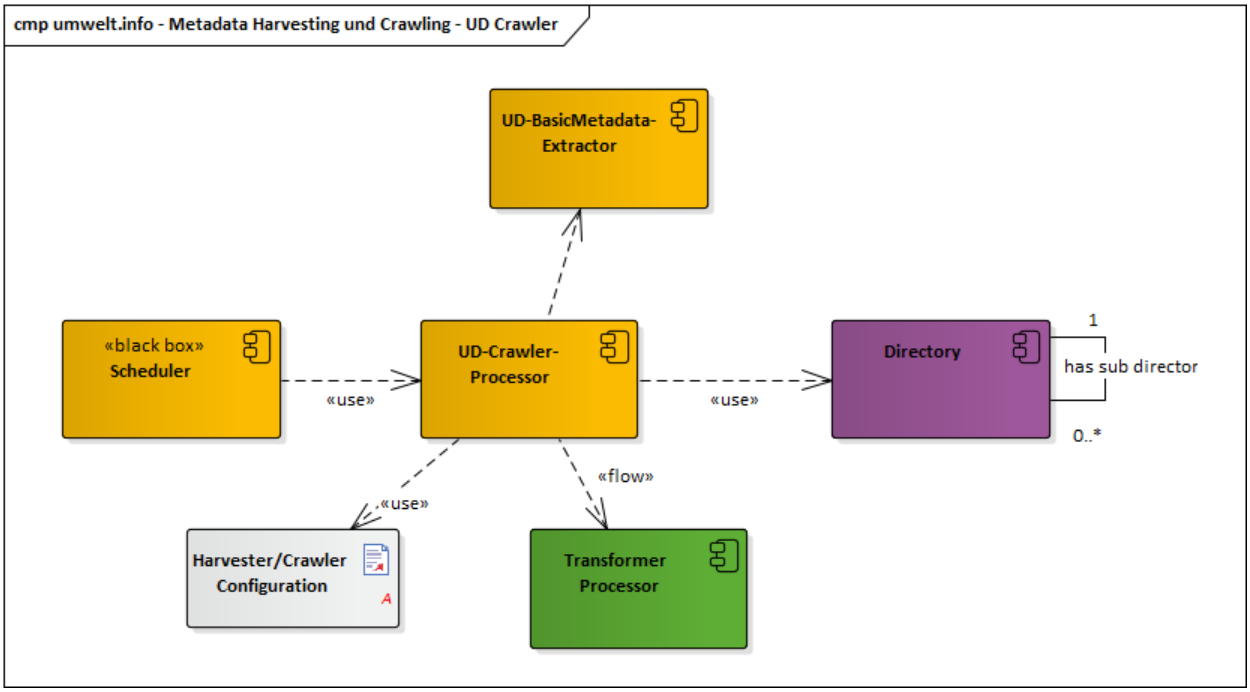
Qualitäts-/Leistungsmerkmale

- Die Crawler sollten durch die Portalbetreiber konfigurierbar sein, so dass etwa die Anfrage an eine zu harvestende Datenquelle (Web-Verzeichnis bzw. ein UNC-Verzeichnis) konfiguriert werden kann.
- Die Schnittstellen sollten idealerweise die Möglichkeit besitzen, lediglich die Service-Metadaten anzufordern, die sich seit dem letzten „Crawling" geändert haben, um nicht immer alle Daten/Metadaten der Daten-Ressource wiederholt crawlen zu müssen.

2.13.6 UD-Crawler (Unstrukturierte Daten)

UD-Crawler sind Crawler von Verzeichnissen mit unstrukturierten Daten.

Abbildung 19: Ebene 3 umwelt.info - Metadata Harvesting und Crawling - UD-Crawler (White Box)



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außen denen die schon zuvor definiert worden sind).

2.13.6.1 UD-Crawler-Processor (Unstrukturierte Daten-Crawler)

Beschreibung (vgl. [2])

Führt das Crawlen durch, weiß wie ein Web-Verzeichnis bzw. ein UNC-Verzeichnis, das er vom Harvest/Crawler-Manager mitgeteilt bekommt, sukzessive zu durchsuchen ist (inklusive Unterverzeichnisse). Der Crawler überlässt die Konvertierung der Metadaten dem UD-BasicMetadata-Extractor und übergibt die Metadaten an den Metadaten-Index.

Weitere Details (vgl. [2])

Ein Unstrukturierte Daten-Crawler dient (wie der Namen bereits sagt) der Ermittlung und Indizierung von unstrukturierten Daten (in Web- oder etwa UNC-Verzeichnissen), etwa Dokumenten (Word oder .pdf wie Berichte, Studien, Informationen zu Öffentlichkeitsarbeit, unstrukturierten Statistiken usw., Präsentationen, Bilder, (Geo-) Daten-Produkte) sowie Portalen / Web-Anwendungen, etwa zu Umweltdaten (z. B. Eignungscheck Windenergie). Der DU-Crawler leitet aus den unstrukturierten Daten automatisch Metainformationen ab, die hier allerdings eher rudimentär ausfallen, da die Inhalte nur sehr generisch behandelt werden können (das (Meta-) Datenmodell ist ja nicht oder nur rudimentär bekannt). Somit sind die Suchmöglichkeiten auf der Basis von Metadaten nach unstrukturierten Daten auch eingeschränkt.

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle, die einen Crawling Auftrag entgegennimmt.

Benötigt:

- Externes Web-Verzeichnis bzw. UNC-Verzeichnis (inklusive Unterverzeichnisse), mit beliebigen Dateien (Word, .pdf, HTML, .ppt, .jpg, .tif, …). Auf diese werden die extrahierten Metadaten verweisen (vgl. [2]).
- Transformer-Dienst, an den die extrahierten Metadaten weitergeleitet werden.

Qualitäts-/Leistungsmerkmale

- Die Crawler sollten durch die Portalbetreiber konfigurierbar sein, so dass etwa die Anfrage an eine zu crawlende Datenquelle konfiguriert werden kann.
- Beim Crawlen sollten idealerweise lediglich die Metadaten abgeholt werden, die sich seit dem letzten „Crawling" geändert haben, um nicht immer alle Metadaten wiederholt harvesten zu müssen.

2.13.6.2 UD-BasicMetadata-Extractor

Beschreibung (vgl. [2])

Leitet aus den Dateien (welche die unstrukturierten Daten enthalten) des Verzeichnisses die grundlegenden Metadaten ab. Dabei kann das interne Metadatenmodell evtl. nicht komplett bedient werden.

Weitere Details (vgl. [2])

Der UD-BasicMetadata-Extractor leitet aus den unstrukturierten Daten automatisch Metainformationen ab, die hier allerdings eher rudimentär ausfallen, da die Inhalte nur sehr generisch behandelt werden können (das (Meta-) Datenmodell ist ja nicht oder nur rudimentär bekannt..

Schnittstelle(n)

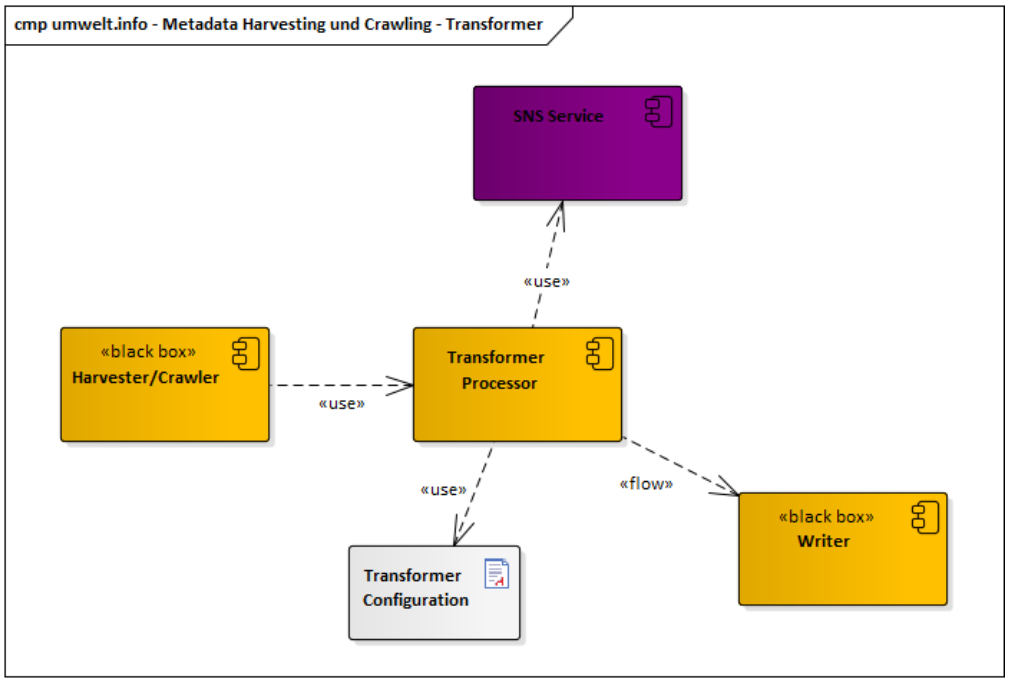
Bereitgestellt:

Schnittstelle, die die unstrukturierten Daten entgegennimmt und die Metadaten zurück liefert.

2.14 Ebene 3 umwelt.info – Metadata Harvesting und Crawling - Transformer (White Box)

Die Transformer übernehmen die Transformation diverser Typen von abgestimmten Metadatenformaten in das Format des internen Metadatenschemas. Sie übernehmen auch das „semantische Anreichern“ der Metadaten. Das Ergebnis der Transformation wird an den Writer übergeben.

Abbildung 20: umwelt.info – Metadata Harvesting und Crawling – Transformer (White Box)



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außen denen die schon zuvor definiert worden sind).

2.14.1 Transfomer Bausteine

2.14.1.1 Transformer Processor

Beschreibung

Die Transformer Prozessoren übernehmen die Transformation diverser definierter Eingabeformate in das Format des internen Metadatenschemas (etwa anhand eines XSL-Stylesheets).

Weitere Details

Es werden zudem auch spezifische Qualitätsverbesserungen vorgenommen, die unabhängig von der Datenquelle sind, etwa die Übersetzung bestimmter Schlüsselworte (z. B. keywords) oder einzelner Wörter des Volltextes in die Begriffswelt einer gemeinsamen Fachsprache, etwa über den Semantischen Netzwerk Service (SNS). Die konvertierten Metadaten werden an den Writer übergeben.

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle, über die die zu transformierenden Metadaten übergeben werden. Letztere können je nach Harvester sehr unterschiedlich aussehen.

Benötigt:

- SNS-Service

Offene Punkte/Probleme/Risiken

Risko: Metadaten nicht konform zum definierten Format.

2.14.1.2 Transformer Configuration

Beschreibung

Die Transformer Configuration enthält Konfigurationsinformationen für den Transformer.

Weitere Details

Bei der Konfiguration kann es sich etwa um ein XSLT-Dokument für das Konvertieren des ankommenden Metadaten-Dokumentes in das interne Metadatenschema handeln.

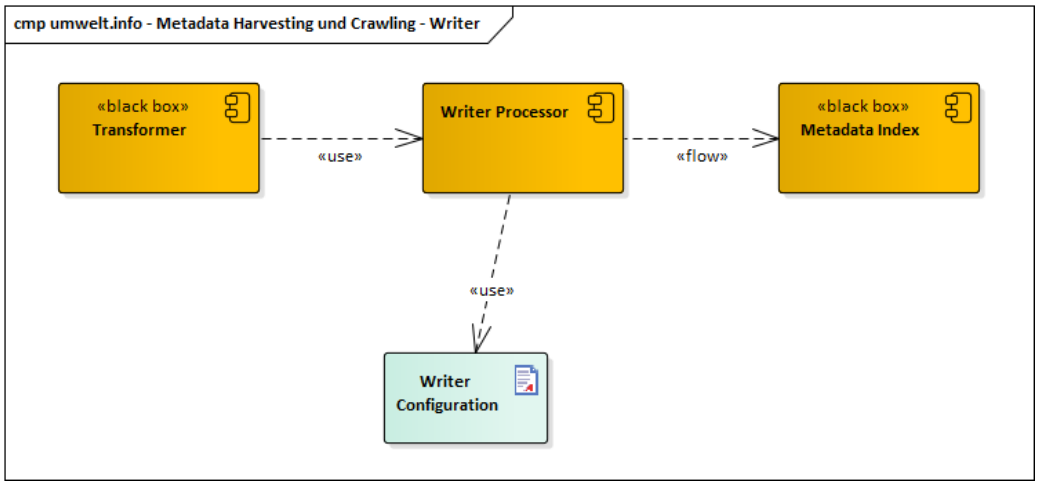
Offene Punkte/Probleme/Risiken

Offen: Benötigte Konfigurationsinformationen

2.15 Ebene 3 umwelt.info – Metadata Harvesting und Crawling- Writer (White Box)

Der Writer veranlasst Bereinigungen der Metadaten im Metadaten-Index und überträgt die Daten dorthin.

Abbildung 21: umwelt.info – Metadata Harvesting und Crawling – Writer (White Box)



Quelle: eigene Darstellung, con terra GmbH

Im Folgenden werden die enthaltenen Bausteine beschrieben (außen denen die schon zuvor definiert worden sind).

2.15.1 Writer Bausteine

2.15.1.1 Writer Processor

Beschreibung

Der Writer Processor führt Bereinigungen der Daten im Metadaten-Index durch und überträgt die Metadaten dorthin, wo sie letztlich auf der Basis bestimmter Elemente indiziert und gespeichert werden.

Weitere Details

Der Writer löscht etwa auch die Datensätze im Metadaten-Index, die in der Datenquelle nicht mehr vorhanden sind. Die Liste aller Identifier aller Metadatensätze bekommt er innerhalb der Prozesskette mitgeteilt.

Vom Metadaten-Index sollte für die ermittelten Dokumente zusätzlich eine Volltextindizierung veranlasst werden.

Für unstrukturierte Daten und Informationen kann die Volltextindizierung die einzige Möglichkeit der Indizierung sein, falls sich von den Harvestern und Crawlern keine Metadaten extrahieren lassen, die das interne Metadatenmodell befriedigen (vgl. [2]).

Schnittstelle(n)

Bereitgestellt:

- Schnittstelle, über die die zu indizierenden Daten (im internen Metadatenschema) übergeben werden.

Benötigt:

- Metadaten (RDF) Change(C)-Update(U)-Delete(D) Schnittstelle des Metadaten-Index

2.15.1.2 Writer Configuration

Beschreibung

Die Writer Configuration enthält Konfigurationsinformationen für den Writer.

Weitere Details

Bei der Konfiguration kann es sich etwa um Informationen zum Zugriff auf den Metadaten-Index (z. B. Endpoint, Index, …) handeln.

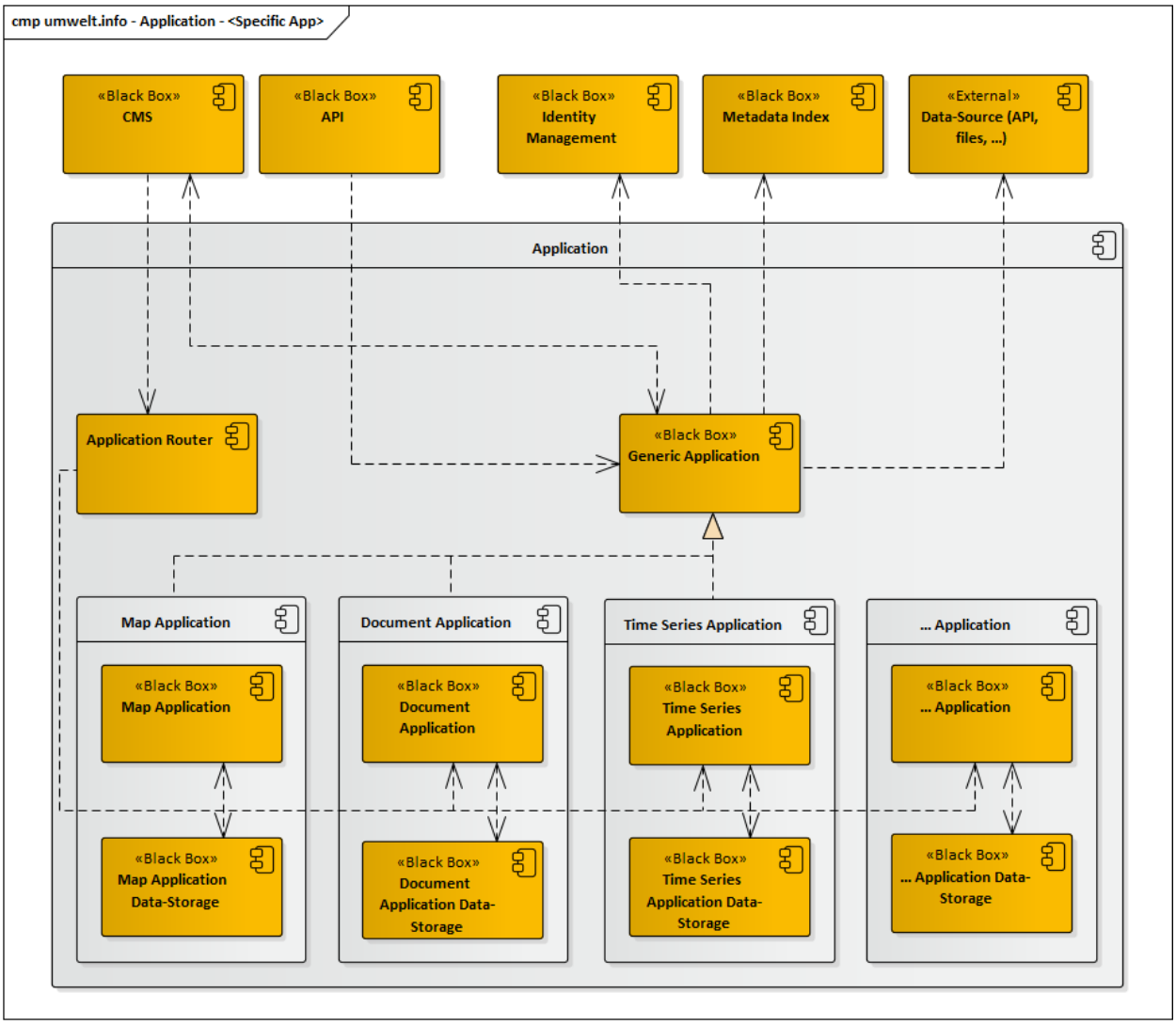
Offene Punkte/Probleme/Risiken

Offen: Benötigte Konfigurationsinformationen

2.16 Ebene 3 umwelt.info – Application – Map/Document/Time Series/… Application (White Box)

Über die Data-Storage Komponenten der unterschiedlichen Anwendungen haben die Nutzer*innen die Möglichkeit, eigene Daten hinzuzuladen und diese mit den gefundenen Daten im umwelt.info Portal miteinander zu visualisieren. Des Weiteren können die hinzugeladenen Daten in den Metadatenkatalog des umwelt.info Portals aufgenommen werden. Dafür muss eine Metadatenerfassung erfolgen, die Nutzer*innen mit der Rolle Datenbereitsteller*in durchführen können.

Abbildung 22: Ebene 3 umwelt.info - Application - Map/Document/Time Series/... Application (White Box)



Quelle: eigene Darstellung, con terra GmbH

2.16.1 Map/Document/Time Series Application Data-Storage

Beschreibung

Der Application Data-Storage dient den Nutzer*innen für die Persistierung eigener Daten, welche in den verschiedenen Anwendungen hinzugeladen werden können.

Weitere Details

Für die Verwendung der Funktion zum Hinzuladen (Import) eigener Daten in den Anwendungen, ist eine Registrierung am umwelt.info Portal notwendig. Beim Hinzuladen der eigenen Daten werden diese im Nutzerprofil der Nutzer*innen hinterlegt. Die Nutzer*innen können entscheiden, ob sie ihre eigenen Daten in das umwelt.info Portal zur Verfügung stellen möchten. Dafür muss eine Metadatenerfassung erfolgen, für die sie berechtigt sein müssen.

Schnittstelle(n)

Bereitgestellt:

- Verschiedene Data-Storage der Anwendungen in der Application

Benötigt:

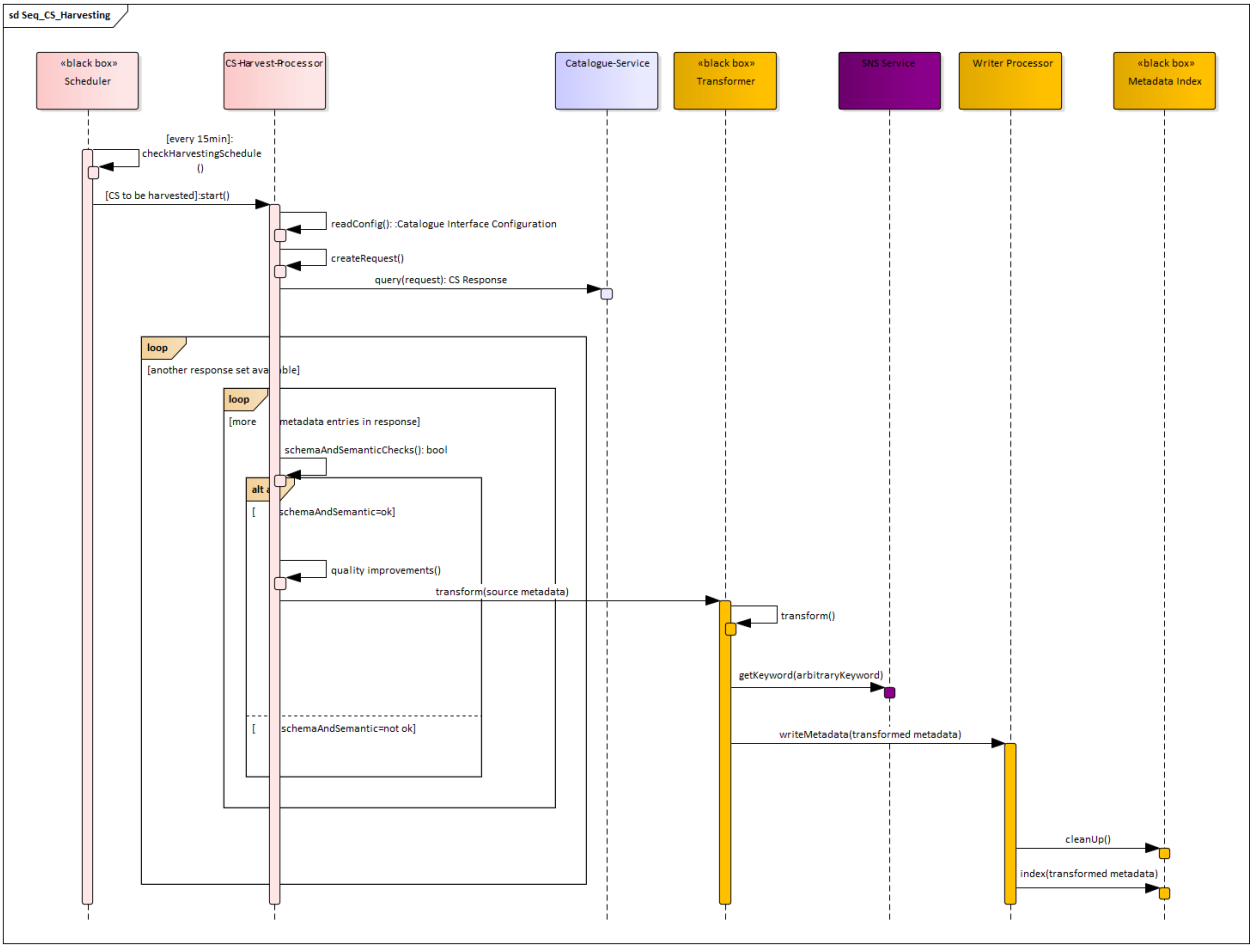
- Metadaten-Index
- Data Check-In
- Identity Management

3 Laufzeitsicht

3.1 Harvesting

Das folgende Sequenzdiagramm zeigt das Zusammenspiel der Komponenten für das Harvesting eines Metadatenkatalogs über eine (bekannte) Catalogue-Service Schnittstelle mittels eines CS-Harvesters.

Abbildung 23: Sequenzdiagramm Harvesting



Quelle: eigene Darstellung, con terra GmbH

3.2 Zielorientierte Suche

Abbildung 24 zeigt ein Sequenzdiagramm, welches das Zusammenspiel der Komponenten bei der zielorientierten Suche zeigt. Die Nutzer*innen gelangen über das CMS zur „Search Website“, die einen Suchschlitz (zur Texteingabe) bereitstellt und über die, nach Eingabe eines oder mehrerer Begriffe, eine Suche durchgeführt werden kann. Dazu wird zunächst über das CMS die „Suchschlitz UI“ angefragt und geladen. Die UI selbst zeigt die Suchhistorie, sobald der Suchschlitz den Fokus erhält (z. B., wenn der Nutzende in den Suchschlitz klickt). Während der Eingabe (etwa ab dem dritten Buchstaben) stellt die UI-Ergebnisvorschläge dar. Dazu kann die UI einerseits im Browser gespeicherte Suchergebnisse (zum Beispiel aus früheren Suchen, Favoriten, etc.) nutzen und

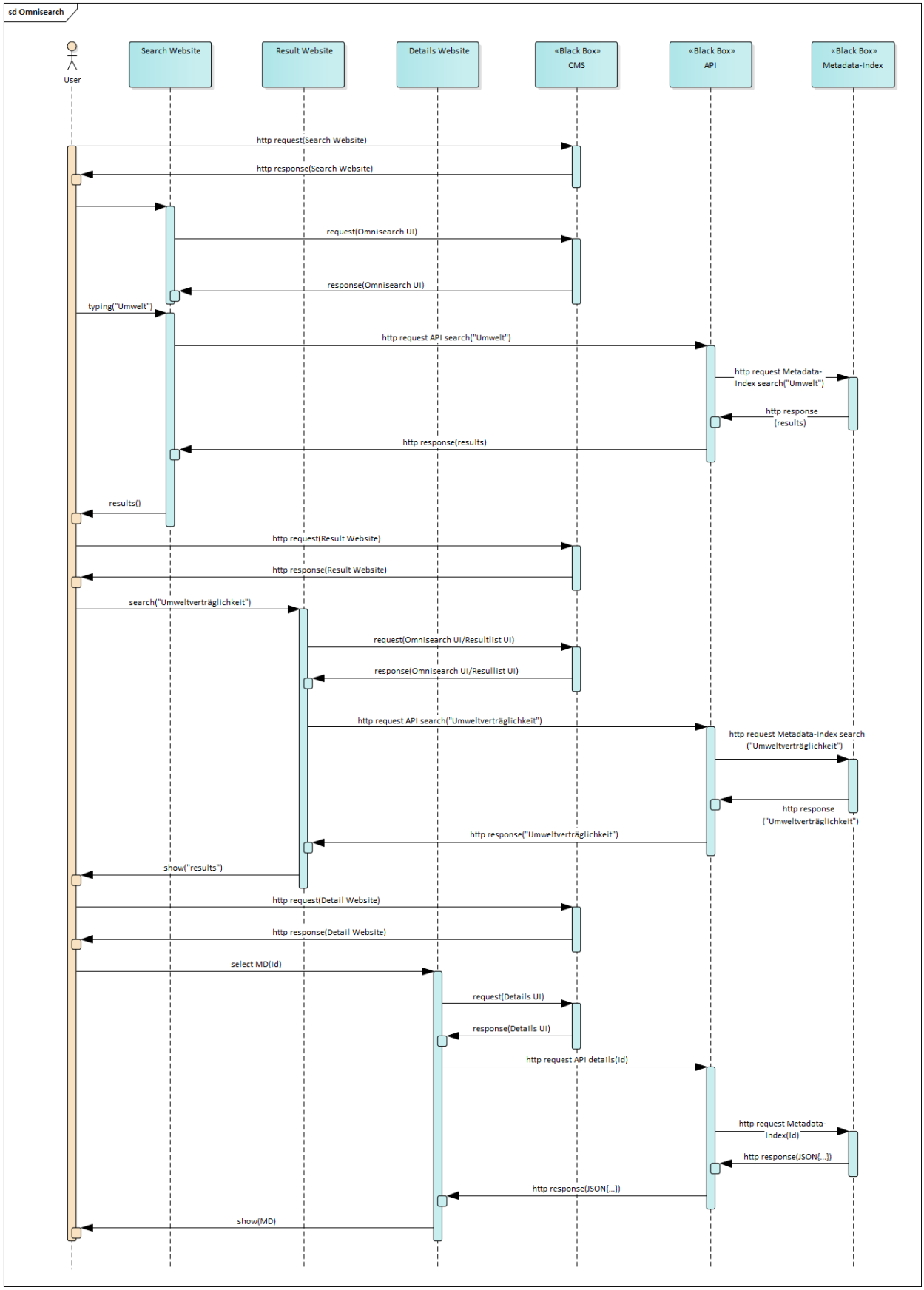
andererseits über den Metadaten-Index Suchergebnisse erhalten. Im weiteren Verlauf der Sucheingabe, zu sinnvollen Zeitpunkten[3] aktualisiert die UI die Vorschau durch neue Suchanfragen. Zum Beispiel kann bei jeder Vervollständigung eines Wortes erneut gesucht werden, um die integrierte Ergebnisvorschau zu aktualisieren.

Vervollständigen die Nutzer*innen ihre Suche und senden diese ab, liefert der Metadaten-Index Ergebnisse, die den Nutzer*innen auf der „Result Website“ angezeigt wird. Durch die Auswahl eines Suchergebnisses gelangen die Nutzer*innen direkt auf die „Detail Website“, welche die Metadaten anzeigt.

Die Nutzung der Portal-Funktionen wird durch „Einbinden“ der UI-Komponenten in den Webseiten des CMS ermöglicht. Dabei werden die UI-Komponenten als JavaScript eingebettet und die UI nachgelagert, d. h. erst im Browser dynamisch gerendert (Suchschlitz UI, Ergebnisseite UI, Detailseite UI). Alternativ kann die Darstellung der UI-Komponenten über das HTML der Websites im CMS realisiert werden. Dabei wird der zugehörige HTML-Code der UI-Komponenten bereits serverseitig gerendert.

[3] Bei der hier dargestellten "integrierten Suchergebnisvorschau" muss für jede Aktualisierung eine Suche an den Server gesendet werden. Das ist aber nicht für jeden neu eingegebenen Buchstaben sinnvoll. Ähnliche Fälle sind eine Wortvervollständigung oder eine Autokorrektur. Diese Funktionen lassen sich zwar mit der integrierten Suche kombinieren, für eine Vorhersage auf Wortbasis muss allerdings nach jedem neuen Buchstaben geprüft werden, ob neue Korrekturen oder Vorschläge abgefragt werden müssen. Daher werden hier keine genauen Angaben gemacht, zu welchen genauen Zeitpunkten neue Vorschläge abgefragt werden müssen.

Abbildung 24: Sequenzdiagramm zielorientierte Suche



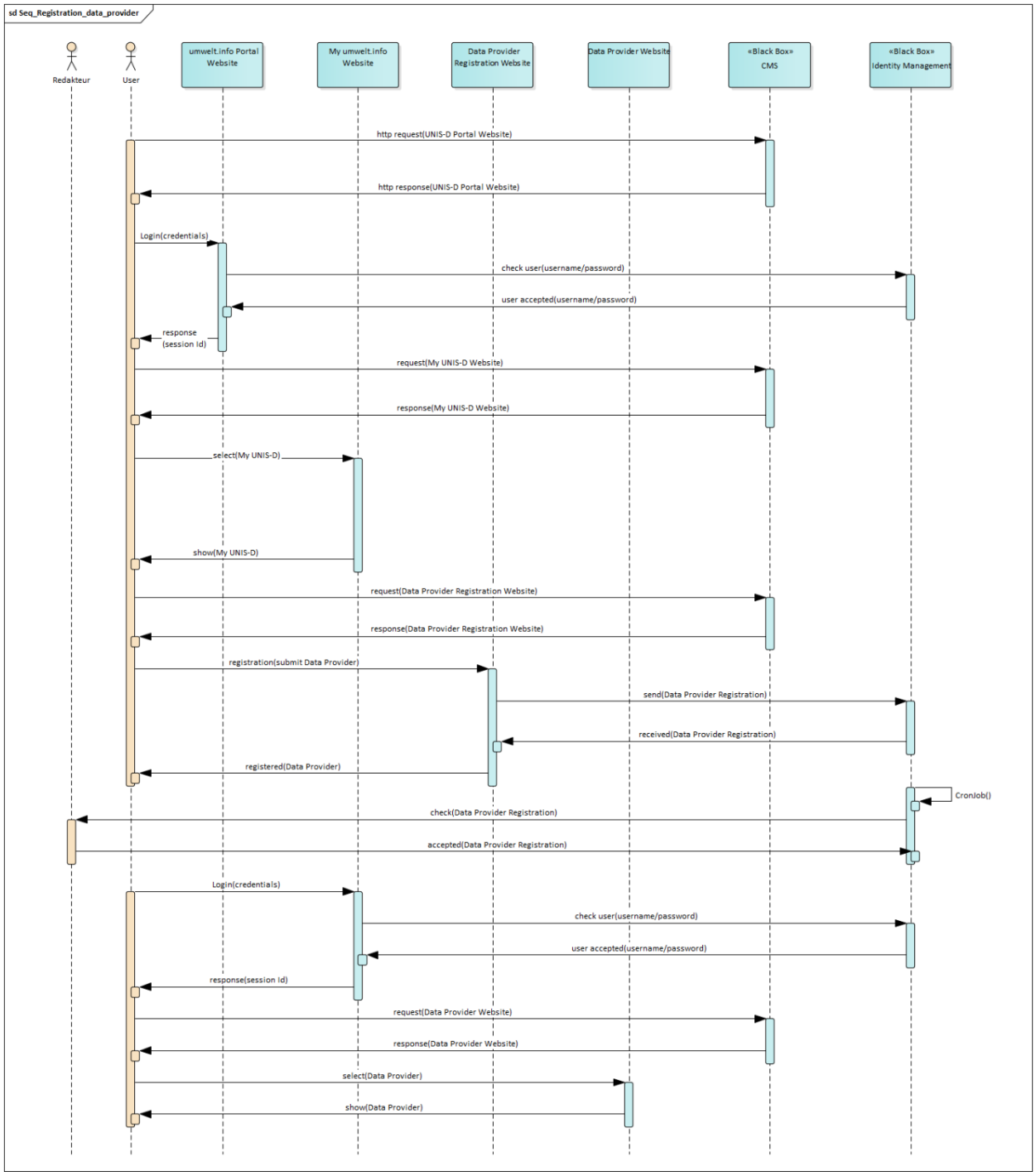
Quelle: eigene Darstellung, con terra GmbH

3.3 Registrieren als Datenbereitsteller*innen

Abbildung 25 zeigt das Zusammenspiel der Komponenten für die Registrierung als Datenbereitsteller*innen. Die Nutzer*innen erreichen über das CMS die „umwelt.info Portal Website“. In dieser erfolgt der Login Vorgang der Nutzer*innen (für ein Login muss bereits vorher eine Registrierung im umwelt.info Portal erfolgt sein). Nach der Anmeldung am umwelt.info Portal, gelangen die Nutzer*innen über den Link „My umwelt.info“ auf die „My umwelt.info Website“. Hier werden den Nutzer*innen die einzelnen Informationen ihres Profils angezeigt. Über die Auswahl „Registrierung als Datenbereitsteller*in“ gelangen die Nutzer*innen auf die „Data Provider Registration Website“. Es erfolgt eine Registrierung als Data Provider, in der die Anfrage an das „Identity Management“ gesendet wird. Über einen „CronJob“ werden in einem festen zeitlichen Intervall die Anfragen zur Registrierung als Data Provider an den „Redakteur“ gesendet, der diese Anfragen prüft und freischaltet. Danach sind die Nutzer*innen als Datenbereitsteller*innen im umwelt.info Portal registriert.

Nach Freischaltung und erneuter Anmeldung am umwelt.info Portal, wird den Nutzer*innen unter der „My umwelt.info Website" die Rolle als Datenbereitsteller*in angezeigt. Damit haben die Nutzer*innen die Möglichkeit, Daten bereitzustellen.

Abbildung 25: Sequenzdiagramm Registrieren als Datenbereitsteller*innen



Quelle: eigene Darstellung, con terra GmbH

4 Verteilungssicht

umwelt.info sollte in einer state-of-the-art IT-Infrastruktur entwickelt und betrieben werden, um eine zukunftssichere Grundlage für den Betrieb und die Weiterentwicklung zu erhalten. Dazu müssen die verschiedenen Komponenten separat und möglichst einfach in den verschiedenen Umgebungen entwickelt, getestet und betrieben werden können. Um diese Ziele zu erreichen, spielen in dieser Architektur „Microservices“ eine entscheidende Rolle. Durch die Abbildung der Komponenten (vgl. Kapitel 2) auf Microservices (vgl. Kapitel 5.2.2) werden die Einzelkomponenten und damit das Gesamtsystem skalierbar und die Pflege und das Management des Systems erheblich vereinfacht.

4.1 Entwicklungs-, Test- und Betriebsumgebungen

Für die verschiedenen Phasen des Softwarelebenszyklus müssen mehrere Instanzen der im Folgenden beschriebenen Infrastruktur existieren.

Dabei ist es für die Entwicklungsumgebung wichtig, dass Änderungen an ihrer Konfiguration möglich sind und die Entwickler*innen diese jederzeit ändern und an ihre Bedürfnisse anpassen können. Diese Umgebung muss während der Entwicklung verfügbar sein. Das Entwicklungssystem stellt dabei keine besonderen Anforderungen an die Verfügbarkeit oder Skalierbarkeit.

Die Testumgebung hilft Änderungen am Produktivsystem vorzubereiten. Um sicherstellen zu können, dass Änderungen am System keine negativen Auswirkungen haben, muss das Testsystem genauso konfiguriert werden wie das Produktivsystem. Änderungen müssen dabei einem formalen, nachvollziehbaren Prozess folgen. Das Testsystem muss für Testzwecke schnell einsatzbereit, aber nicht dauerhaft verfügbar sein.

Die Produktivumgebung muss hochverfügbar sein und zum Beispiel hohe Lasten durch hohe Aktivität der Nutzer*innen verarbeiten können. Dabei müssen Änderungen einem formalen, nachvollziehbaren Prozess folgen, der auch das Deployment und die Durchführung von Tests auf dem Testsystem vorsieht.

4.2 Infrastruktur Ebene 1 Gesamtsystem

Für umwelt.info müssen (virtuelle) Maschinen, in einer Cloud-Umgebung oder in einem eigenen Rechenzentrum betrieben werden. Diese Maschinen dienen als Host-Infrastruktur, welches als Cluster (Abbildung 26) zusammengefasst wird und in dem die vorhandenen Ressourcen flexibel geteilt werden können. In diesem Cluster laufen dann die containerisierten Komponenten.

Für das Deployment der Komponenten im Cluster soll ein standardisierter, automatisierter Prozess eingerichtet werden (vgl. Kapitel 5.3.3, Automatisiertes Delivery / Deployment (CI/CD)). Dazu werden die Komponenten als Container-Images [18] entwickelt und bereitgestellt. Containerisierte Komponenten verfügen über Eigenschaften, die die Entwicklung verteilter, Microservice-basierter Systeme vereinfachen (vgl. Kapitel 5.4.3).

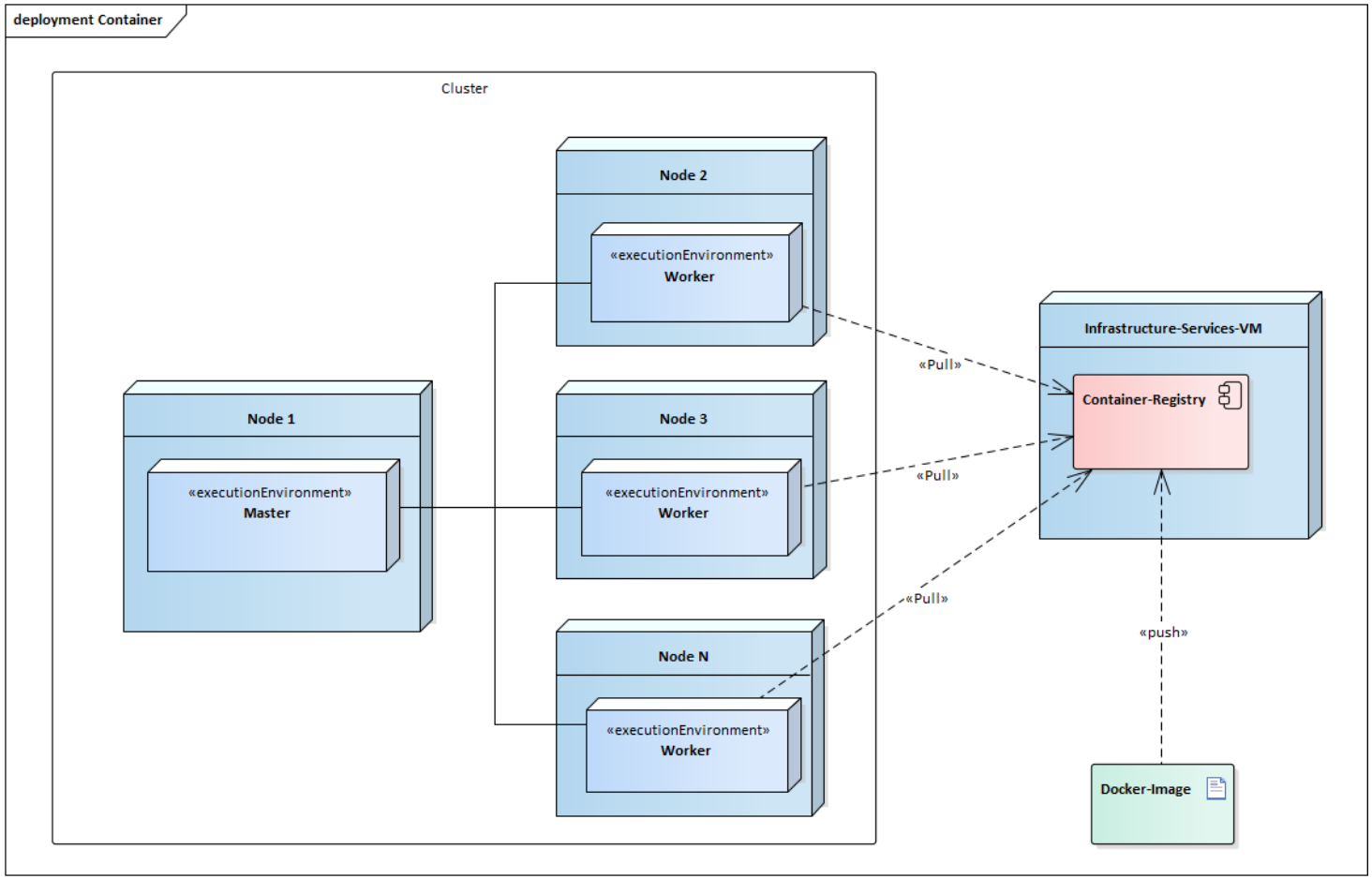
Die Verteilung erfolgt dann als Bereitstellung der Docker-Images über eine Container-Registry. Die Container-Registry verwaltet und versioniert die bereitgestellten Images.

Eine Option für den Aufbau eines solchen Clusters ist Kubernetes [19]. Kubernetes ist eine Orchestrierungsplattform zur Ausführung von Container-Images in einem Cluster von (virtuellen) Maschinen, die hier als „Nodes“ bezeichnet werden.

In Kubernetes wird deklarativ ein Gesamtzustand des Clusters konfiguriert, mit dem sich z. B. die verfügbaren Ressourcen wie Central Processing Unit (CPU) und Speicher zuteilen lassen. Verantwortlich für die Herstellung dieses Zustands ist die Kubernetes Control Plane. Die Control Plane wird in der Regel auf einer oder mehreren separaten Master Maschinen konfiguriert.

Die Container-Images werden auf den Worker Nodes des Clusters ausgeführt. Kubernetes verwaltet Container in Pods. Ein Pod enthält immer einen oder mehrere zusammengehörige Container. Die Control Plane sorgt dafür, dass die konfigurierte Anzahl an Pods mit den dafür notwendigen Systemressourcen auf den Worker Nodes des Clusters ausgeführt werden. Dabei wird das Container-Image von der Container Registry abgerufen and auf dem Worker gestartet.

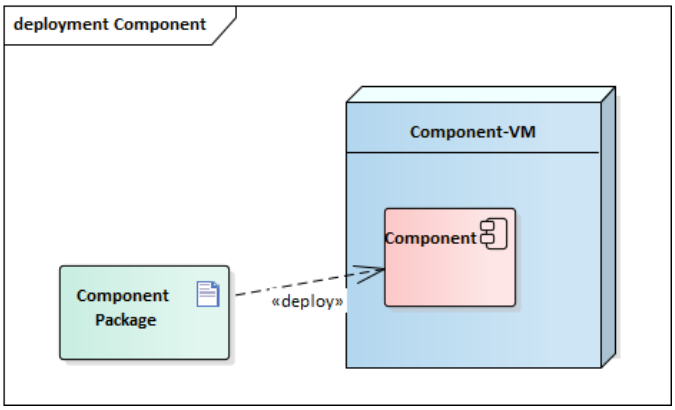
Abbildung 26: Container Deployment



Quelle: eigene Darstellung, con terra GmbH

Falls notwendig werden einzelne Systemteile, die nicht containerisiert werden können, auf eigenen Maschinen installiert. Für diese gelten dann möglicherweise abweichende, nicht automatisierte und standardisierte Vorgehen bei der Verteilung, die durch die jeweiligen Hersteller bestimmt werden (Abbildung 27). Nach Möglichkeit sollten solche Komponenten dennoch so konfiguriert werden, dass der Zugriff durch die Nutzer*innen von außen einheitlich und genau wie der Zugriff auf die anderen Komponenten, über einen Reverse Proxy und über das API-Gateway erfolgt (vgl. 5.2.3).

Abbildung 27: Direktes Deployment einer Komponente



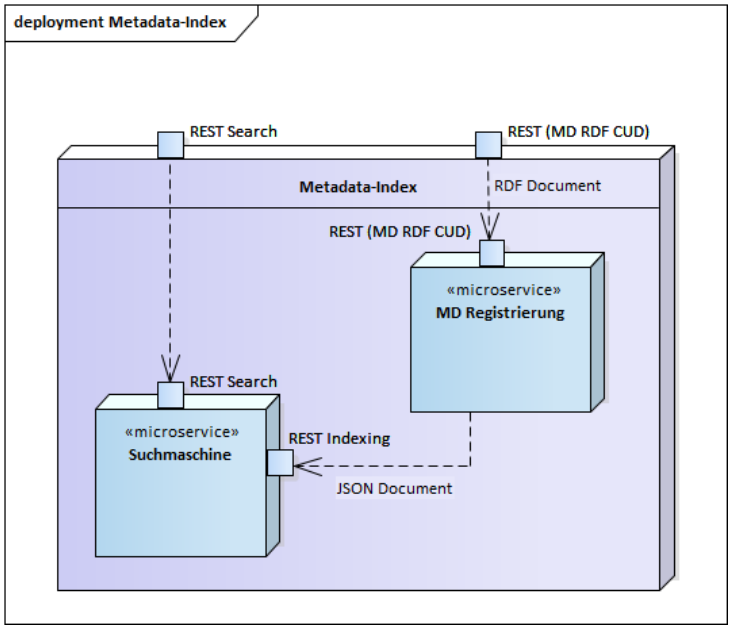
Quelle: eigene Darstellung, con terra GmbH

Auf der folgenden Ebene 2 werden Microservices dargestellt, die jeweils das Deployment einer Ebene 1-Komponente darstellen. Dabei werden White Boxen aus der Komponentensicht als Microservice ausgewählt, wenn diese hinreichend feingranular beschrieben wurden und eine weitere Zerlegung keine weitere Teilung von Verantwortlichkeiten bewirkt. Dieses Vorgehen wird zunächst exemplarisch für den Metadaten-Index (Kapitel 4.3) dargestellt und gilt für die meisten anderen Komponenten analog. Eine Besonderheit gilt für das Harvesting Subsystem (Kapitel 4.4), bei dem die Komposition von Microservices für das Harvesten durch Workflows definiert wird.

4.3 Infrastruktur Ebene 2 Metadaten-Index

Der Metadaten-Index (Abbildung 28) repräsentiert das Subsystem für das Speichern, Indizieren und Suchen von Metadaten (vgl. Kapitel 2.1.4). Er besteht (in der nicht linked Data fähigen Lösung) aus einem Microservice zur Metadaten Registrierung und einem Microservice zur Metadaten Indizierung und Suche.

Abbildung 28: Der Metadaten-Index basierend auf 2 Microservices



Quelle: eigene Darstellung, con terra GmbH

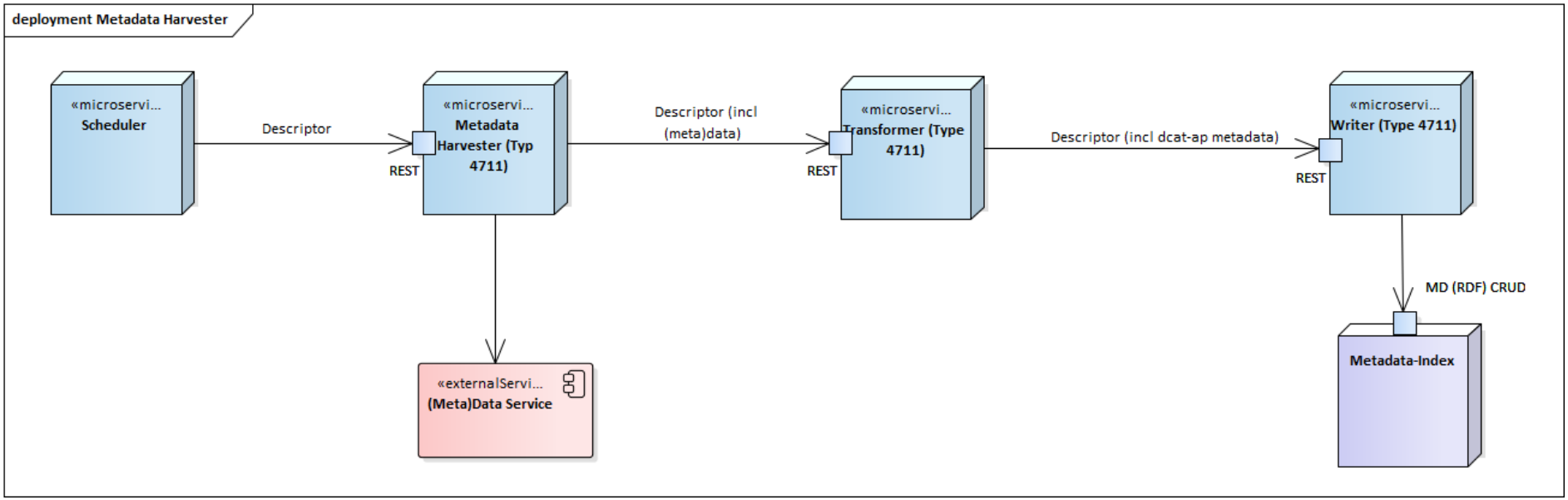
4.4 Infrastruktur Ebene 2 Harvesting-Subsystem

Die zum Harvesting-Subsystem (vgl. Kapitel 2.1.2) gehörenden Services stellen jeweils REST-Interfaces bereit, worüber die Services in einer generischen Art verbunden und orchestriert werden können. Hiermit werden Prozessketten (Workflows) realisiert (Abbildung 29). Workflows benötigen dabei keine zentrale Instanz für die Orchestrierung der Dienste.

Ein Workflow ist durch einen Deskriptor beschrieben, wobei jeder Service den für ihn notwendigen Teil des Deskriptors interpretiert, die nötigen Schritte durchführt und den Deskriptor an den nächsten Service weiterreicht. Jeder Service muss einen Endpunkt bereitstellen, um den Deskriptor zu empfangen.

Die zu prozessierenden Daten können entweder ebenfalls im Deskriptor gespeichert werden oder als Zeiger auf einen Datenspeicher vorliegen.

Abbildung 29: Das Harvesting-Subsystem basierend auf Microservices und Workflow Fähigkeiten



Quelle: eigene Darstellung, con terra GmbH

5 Querschnittliche Konzepte

5.1 Fachliche Konzepte

5.1.1 Metadatenmodell

Ein wesentliches Konzept für die Architektur ist das Metadatenmodell. Dieses spiegelt sich in allen Komponenten: Im Metadaten-Index, in der Suche und den Suchergebnissen, in den Harvestern/Crawlern, in den verschiedensten Transformatoren, im Data-CheckIn, in der *Application* und dem User Interface.

Die Auswahl der Elemente basiert auf den Anforderungen aus dem umwelt.info Projekt und vielfältigen Erfahrungen der con terra mit Metadateninformationssystemen. Außerdem basiert die Auswahl auf Standards wie "Infrastructure for Spatial Information in the European Community" (INSPIRE) Metadatenspezifikationen und "Open Geospatial Consortium" (OGC) Catalogue-Services. Da im umwelt.info Portal unterschiedliche Ressourcentypen indexiert und auffindbar gemacht werden müssen, werden zwei Vokabulare für die Indexierung empfohlen, die parallel unterstützt werden sollten:

1. DCAT-AP.de speziell für Datensätze (dafür wurde Data Catalogue Vocabulary (DCAT) [20] entwickelt)
2. Schema.org für andere Ressourcentypen (z. B. Dokumente, Videos)

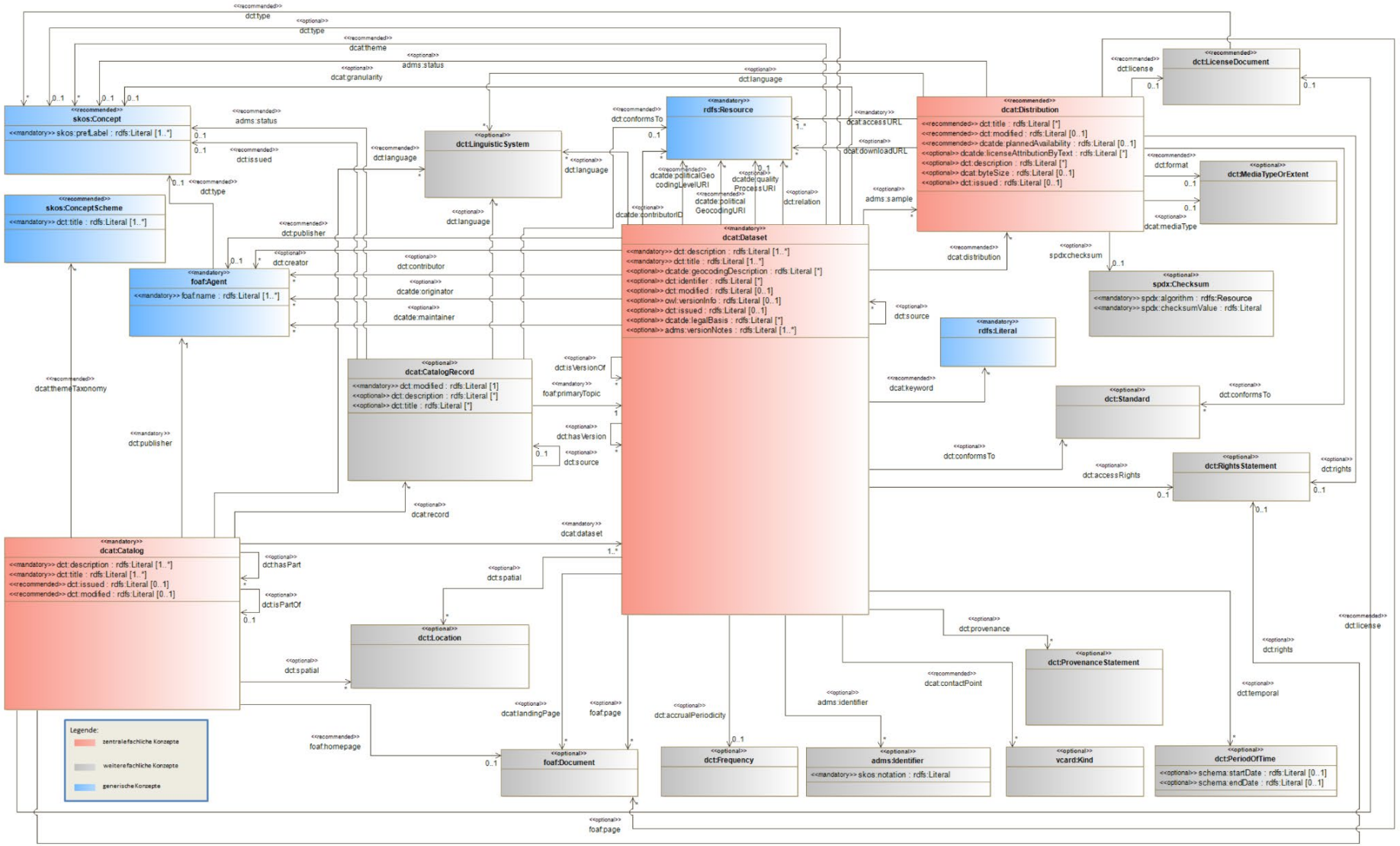
5.1.1.1 DCAT-AP.de

Als Grundlage für die Beschreibung von Datensätzen wird das Metadatenmodell DCAT-AP.de vorgeschlagen [21], ein deutsches Profil (Ableitung) des europäischen Metadatenstandards DCAT-AP (Data Catalogue Vocabulary-Application Profile for data portals in europe) [22]. DCAT-AP.de wurde „im Auftrag des Staatsbetrieb Sächsische Informatik Dienste (SID) erstellt, in die Herausgeberschaft von GovData überführt, im Juni 2017 veröffentlicht und im Juni 2018 vom IT-Planungsrat beschlossen. Sie stellt eine direkte Kompatibilität zum EU-Standard sicher und ist der Vorschlag des Freistaats Sachsen für den Datenaustausch im GovData-Verbund. Die vorliegende Spezifikation wird von der Geschäfts- und Koordinierungsstelle GovData (GKSt) weiter gepflegt und in die Bearbeitung des beim IT-Planungsrat geltend gemachten Standardisierungsbedarfs einfließen. Sie regelt verbindlich, wie Daten auf dezentraler Seite auszuzeichnen und zur zentralen deutschlandweiten Bereitstellung im GovData-Portal anzuliefern sind" ([23], S. 2).
Die wichtigsten Elemente werden im Folgenden dargestellt. Diese lassen sich im Wesentlichen unterscheiden in Elemente, die die Eigenschaften des Datensatzes beschreiben (Klasse: Dataset), in Elemente, die die Meta-Metadaten enthalten, also z. B. das Erstellungsdatum des Metadatensatzes (Klasse: CatalogRecord) und in Elemente, die die Eigenschaften eines Zugriffsverfahrens beschreiben, mit dem auf die Daten des Datasets zugegriffen werden kann (Klasse: Distribution) unterteilen.

Die folgende Abbildung 30 zeigt die Klassen und Beziehungen des DCAT-AP.de Datenmodells[4]

[4] Zum Zeitpunkt der grundlegenden Erstellung dieses Dokumentes (Mitte 2021) war die Version 1.1 aktuell.

Abbildung 30: DCAT-AP.de Metadatenmodell (Version 1.1) [23]



Quelle: DCAT-AP.de Spezifikation. Deutsche Adaption des „Data Catalogue Application Profile“ (DCAT-AP) für Datenportale in Europa, Version: 1.1

In der folgenden Tabelle sind die Elemente dargestellt, jeweils mit:

- Name des Elementes
- Kurze Beschreibung
- DCAT-AP.de Datentyp (der selber auch eine Datenstruktur (z. B. Agent) und nicht nur ein Basistyp (z. B. Literal) sein kann)
- Verbindlichkeitsstufe (0..1, 1, 0..*, 1..*)
- Element im DCAT-AP.de Standard (sofern vorhanden)
- Ist das Element suchbar im Metadaten-Index
- Wird das Element im Metadateneditor angeboten

Entsprechend dem DCAT-AP.de Modell wird die Klasse Dataset mit einem dcat:Dataset, die Klasse CatalogRecord mit dcat:CatalogRecord und die Klasse Distribution mit einer dcat:Distribution Datenstruktur umgesetzt.

Daneben ist eine Spalte vorhanden die anzeigt, ob das Element zur Bearbeitung im Metadaten-Editor vorhanden sein sollte. Die hier vorgesehenen Elemente, die für den MD Editor vorgesehen sind, können bei Bedarf angepasst werden. Diese Anpassungen können zum Beispiel im Rahmen der Umsetzungsphase erfolgen.

Tabelle 1: Klasse: CatalogRecord (DCAT-AP.de: dcat:CatalogRecord) (nach [23])

Element-name (englisch)	Beschreibung	Datatype	Verbindl. in DCAT-AP.de	Element in DCAT-AP.de	MD Editor
Metadata ID	Intern erzeugte ID des Metadatensatzes	Literal/URI			**x** (**obligatorisch**, automatisch erzeugt) Speicherng als dct:identifier
CreationDate Metadata	Wann wurde der Metadatensatz erstellt	DateTime	0..1	dct:issued	**x** (automatisch erzeugt)
ModificationDate Metadata	Wann wurde der Metadatensatz verändert	DateTime	1	dct:modified	**x** (**obligatorisch**, automatisch erzeugt)
Dataset	Der zugehörige Datensatz	dcat:Dataset	1	foaf:primaryTopic	**x** (**obligatorisch) <Tabelle Dataset>**
Sprache	Diese Eigenschaft bezieht sich auf die Sprache der Metadatenbeschreibung für die	dct:LinguisticSystem	0..*	dct:language	**x** (automatisch erzeugt)

Element-name (englisch)	Beschreibung	Datatype	Verbindl. in DCAT-AP.de	Element in DCAT-AP.de	MD Editor
	zum Katalogeintrag gehörenden Datenstrukturen (z. B. Titel, Beschreibungen usw.). Diese Eigenschaft kann wiederholt werden, falls die Metadaten in verschiedenen Sprachen zur Verfügung stehen.				

Tabelle 2: Klasse: Dataset (DCAT-AP.de: dcat:Dataset) (nach [23])

Element-name (englisch)	Beschreibung	Datatype	Verbindl. in DCAT-AP.de	Element in DCAT-AP.de	MD-Editor
ID	Diese Eigenschaft enthält die Haupt-ID der Datenstruktur im Kontext des jeweiligen Kataloges (z. B. die URI-Adresse oder eine andere eindeutige ID).	rdfs:Literal	0..*	dct:identifier	**x (obligatorisch, automatisch erzeugt)**
Title	Titel des Datensatzes	rdfs:Literal	1..*	dct:title	**x (obligatorisch)**
Description	Beschreibung des Datensatzes	rdfs:Literal	1..*	dct:description	**x (obligatorisch)**
Publisher	Wer publiziert den Datensatz	foaf:Agent	0..1	dct:publisher	
Contact	Wer ist der Ansprechpartner für den Datensatz	vcard:Kind	0..*	dcat:ContactPoint	**x**
Originator (Creator)	Urheber des Datensatzes	foaf:Agent	0..*	dcatde:originator	
Maintainer	Diese Eigenschaft verweist auf die Stellen oder Personen, die Verantwortung und Rechenschaftspflicht für die Daten und ihre angemessene Pflege übernehmen.	foaf:Agent	0..*	dcatde:maintainer	
Bearbeiter	Diese Eigenschaft verweist auf Stellen oder Personen, die die Daten bearbeitet haben (z. B.	foaf:Agent	0..*	dcat:contributor	

Element-name (englisch)	Beschreibung	Datatype	Verbindl. in DCAT-AP.de	Element in DCAT-AP.de	MD-Editor
	durch Formatierung derselben).				
Autor	Diese Eigenschaft verweist auf Stellen oder Personen, die die Daten erstellt haben. Die Autorenschaft umfasst für gewöhnlich das Recht am geistigen Eigentum.	foaf:Agent	0..*	dct:creator	
Keyword	Schlagwort zum Datensatz	rdfs:Literal	0..*	dcat:keyword	**x**
Theme	Kategorie (Theme) des Datensatzes. Hier lassen sich über einen entsprechenden Thesaurus etwa Datentypen wie 'Karten', 'Info-Grafiken', 'Web-Seite', 'Schul-Material',... oder aber auch spezielle Qualitätsanforderungenwie 'maschinenlesbar', 'offene Lizenz' etc unterscheiden	skos:Concept Eintrag aus einem "Theme-Vokabular"	0..*	dcat:theme	**x (obligatorisch)**
PoliticalGeocodingURI	Diese Eigenschaft verknüpft eine Datenstruktur mit dem von ihr abgedeckten administrativen Gebiet	rdfs:Resource (Für die Referenzierung sollen die auf http://dcat-ap.de/def/veröffentlichten Wertelisten verwendet werden.)	0..*	dcatde:politicalGeocodingURI	**x**
PoliticalGeocodingLevelURI	Geopolitische Abdeckung der Datenstruktur, etwa durch Kennzeichnung der Verwaltungsebene Bund, Bundesland, Kreis oder Kommune, als dcat-ap.de URI	rdfs:Resource (Für die Referenzierung sollen die auf http://dcat-ap.de/def/veröffentlichten Wertelisten verwendet werden.)	0..*	dcatde:politicalGeocodingLevelURI	**x**

Element-name (englisch)	Beschreibung	Datatype	Verbindl. in DCAT-AP.de	Element in DCAT-AP.de	MD-Editor
CreationDate Data	Diese Eigenschaft enthält das Datum der Herausgabe/Emission (z. B. in Form einer Veröffentlichung) der Datenstruktur.	xsd:dateTime	0..1	dct:issued	x
Modification Date Data	Diese Eigenschaft erfasst das Datum der letzten Aktualisierung bzw. Modifikation der Datenstruktur	xsd:dateTime	0..1	dct:modified	**x**
Distribution	Diese Eigenschaft verknüpft die Datenstruktur mit einer verfügbaren Distribution.	dcat:Distribution	1..*	dcat:distribution	**x**
LandingPage	Diese Eigenschaft verweist auf eine Webseite, welche Zugriff auf die Datenstruktur, ihre Distributionen und/oder weitere Informationen ermöglicht. Es ist beabsichtigt, auf die Webseite des originären Datenbereitstellers zu verweisen und nicht auf zwischengeschaltete Intermediäre.	dcat:landingPage	0..*	foaf:Document	
Spatial Extent	Diese Eigenschaft bezieht sich auf eine geographische Region, welche durch die Datenstruktur abgedeckt wird.	dct:Location Examples: BBox/WK“: "locat”on"“: "BBOX (1000.0, 1002.0, 2000.0, 1000”0)" oder "location" : { "type"“: "polygon", "coordinates" : [[[1000.0, -1001.0], [1001.0, -	0..*	dct:spatial	**x**

Element-name (englisch)	Beschreibung	Datatype	Verbindl. in DCAT-AP.de	Element in DCAT-AP.de	MD-Editor
		1001.0], [1001.0, -1000.0], [1000.0, -1000.0], [1000.0, -1001.0]]] }			
Temporal Extent	Diese Eigenschaft bezieht sich auf eine zeitliche Dauer, welche durch die Datenstruktur abgedeckt wird.	dct:PeriodOfTime	0..*	dct:temporal	
Data ID	Diese Eigenschaft verweist auf sekundäre IDs der Datenstruktur wie beispielsweise DOI	adms:Identifier	0..*	adms:identifier	
Type	Typ der Datenstruktur. Zur Gruppierung von linearen und nicht- linearen Reihen/ Collections ist gemäß dem DCAT-AP.de Konventionenhandbuch eine Gruppenstruktur vom Typ „Collection“ anzulegen, die auf die gruppierten Daten- strukturen mittels „Weitere Version“ (dct:hasVersion) verweist. Alle gruppierten Datenstrukturen verweisen dann mittels „ist Version von“ (dct:isVersionOf) auf die URI (http://dcat-ap.de/def/datasetTypes/collection) dieser logischen Klammer. Das ist in ISO19115 die Beziehung “series” (collection) - “dataset”.	skos:Concept	0..1	dct:type	
Version	Diese Eigenschaft enthält eine Versionsnummer oder anderweitige Versionskennzeichnung der Datenstruktur.	rdfs:Literal	0..1	owl:versionInfo	

Element-name (englisch)	Beschreibung	Datatype	Verbindl. in DCAT-AP.de	Element in DCAT-AP.de	MD-Editor
LegalBasis	Dieses Feld dokumentiert als Freitext optional die Rechtsgrundlage für den Zugang zu den Informationen (die Zugangseröffnung), d. h. die originäre Rechtsgrundlage für den Zugang zu Daten der Verwaltung. z. B. Public Sector Information Directive (PSI-Direktive), Umweltinformationsgesetz (UIG), deutsche Informationsfreiheits- (IFG) und Transparenzgesetze. Diese Eigenschaft kann für parallele Sprachversionen wiederholt werden.	rdfs:Literal	0..*	dcatde:legalBasis	
QualityProcessURI	Eine URI, die auf den Prozess zur Qualitätssicherung der Datenstrukturen verweist. Es handelt sich idealerweise um die URL einer Webseite.	rdfs:Resource	0..1	dcatde:qualityProcessURI	
GeocodingDescription	Geografische Abdeckung repräsentiert durch die Bezeichnung eines administrativen Gebiets oder eines fachlichen Bezugs als Freitext. Ergänzend als Text bzw. alleinstehend für alle Fälle bei denen die geopolitische Abdeckung nicht durch eine URI angegeben werden kann (z. B. bei komplexeren Bund-Länder-Kooperationen oder auf kommunaler Ebene).	rdfs:Literal	0..*	dcatde:geocodingDescription	**x**
Granularity	Diese Eigenschaft beschreibt die durch die Datenstruktur abgedeckte zeitliche Granularität (z. B. wöchentlich, monatlich, jährlich).	skos:Concept	0..1	dcat:granularity	
AccessRights	Diese Eigenschaft verweist auf Informationen, die darlegen, ob die Datenstruktur öffentlich zugänglich ist, Zugriffseinschränkungen	dct:RightsStatement	0..1	dct:accessRights	

Element-name (englisch)	Beschreibung	Datatype	Verbindl. in DCAT-AP.de	Element in DCAT-AP.de	MD-Editor
	existieren oder nicht-öffentlich ist.				

Tabelle 3: Klasse: Distribution (DCAT-AP.de: dcat:Distribution) (nach [23])

Elementname (englisch)	Beschreibung	Datatype	Verbindl.	Element in DCAT-AP.de	MD-Editor
Description	Diese Eigenschaft enthält eine Freitextbeschreibung der Distribution.	rdfs:Literal	0..*	dct:description	**x**
AccessURL	URL-Adresse, die Zugriff auf die Distribution einer Datenstruktur ermöglicht. Die mit der Zugangs-URL erreichbare Ressource kann Informationen zur Verfügung stellen, wie die Distribution erreicht werden kann (landingPage, API, etc.)[5].	rdfs:Resource	1..*	dcat:accessURL	**x (obligatorisch)**
Download-URL	Diese Eigenschaft enthält eine URL-Adresse, welche einen direkten Zugriff/Link auf die herunterladbare Datei im beschriebenen Format liefert.	rdfs:Resource	0..*	dcat:downloadURL	**x**
License (s. 5.1.1.3)	Lizenz, unter welcher die Distribution zur Verfügung gestellt wird.	dct:LicenseDocument	0..1	dct:license	x
Media Type	Diese Eigenschaft bezieht sich auf den Medientyp der Distribution gemäß des von IANA definierten und zur Verfügung gestellten offiziellen Medientypregisters.	dct:MediaTypeOrExtent	0..1	dcat:mediaType, subproperty of dct:format	**x (obligatorisch, wenn nicht Format)**

[5] If the distribution(s) are accessible only through a landing page (i.e. direct download URLs are not known), then the landing page link SHOULD be duplicated as dcat:accessURL on a distribution.

Elementname (englisch)	Beschreibung	Datatype	Verbindl.	Element in DCAT-AP.de	MD-Editor
	Beispiele könnten IANA Media Typen sein für PDF, WMS, WCS, Shapefile, …				
Format	Diese Eigenschaft verweist auf das Datenformat der Distribution im Falle, dass kein offizieller IANA Medientyp (s. Media Type) existiert	dct:MediaTypeOrExtent	0..1	dct:format	**x (obligatorisch, wenn nicht Media Type)**
Size in Bytes	Diese Eigenschaft enthält die Größe der Distribution in Bytes.	xsd:decimal	0..1	dcat:byteSize	
Checksum	Diese Eigenschaft stellt einen Mechanismus zur Verfügung, mit dem sichergestellt werden kann, dass die Inhalte der Distribution sich nicht verändert haben.	spdx:Checksum	0..1	spdx:checksum	
Documentation	Diese Eigenschaft verweist auf eine Webseite oder ein Dokument (enthält eine URL-Adresse) mit Informationen über die Distribution.	foaf:Document	0..*	foaf:page	
Application Profile of the data	Diese Eigenschaft verweist auf eine eingehaltene Regelkonfomität, auf Konformität der Datenstruktur zu einem anderen Standard der Version eines Applikationsprofils der Datenstruktur.	dct:Standard	0..*	dct:conformsTo	
Publication date	Diese Eigenschaft enthält das Datum der Herausgabe/Emission (z. B. in Form einer Veröffentlichung) der Distribution.	xsd:dateTime	0..1	dct:issued	
Distribution rights	Diese Eigenschaft verweist auf eine juristische Quelle, welche die mit der	dct:RightsStatement	0..1	dct:rights	

Elementname (englisch)	Beschreibung	Datatype	Verbindl.	Element in DCAT-AP.de	MD-Editor
	Distribution assoziierten Rechte spezifiziert.				
Status	Diese Eigenschaft bezieht sich auf den Status/Reifegrad der Distribution. Es MUSS das ADMS-Vokabular (http://purl.org/adms/status/1.0) verwendet werden.	skos:Concept	0..1	adms:status	
Title	Diese Eigenschaft bezeichnet den einer Distribution zugewiesenen Titel. Diese Eigenschaft kann für parallele Sprachversionen des Distributionstitels wiederholt werden.	rdfs:Literal	0..*	dct:title	**x (obligatorisch)**
ModificationDate	Diese Eigenschaft erfasst das Datum der letzten Aktualisierung bzw. Modifikation der Distribution	xsd:dateTime	0..1	dct:modified	**x**
Verfügbarkeit	Verfügbarkeit der Distribution einer Datenstruktur als Auswahl aus einer festen Liste von Werten via DCAT-AP URIs.	rdfs:Resource	0..1	dcatde:plannedAvailability	
Namensnennungstext für „By“-Clauses	Hilfskonstrukt zur Speicherung von verpflichtenden Namensnennungstexten aus Lizenzangaben, bis zur Lösung in DCAT-AP.	rdfs:Literal	0..*	dcatde:licenseAttributionByText	

Die folgende Tabelle zeigt Beispiele für Distributionen (nur mit den wichtigsten Eigenschaften):

Tabelle 4: Beispieldistributionen

Titel	AccessURL	Download-URL	MediaType	Format
OGC WMS (Service	https://geo.woudc.org/ows?service=WMS&version=1.3.0&request=	-	application/vnd.ogc.wms_xml	-

Titel	AccessURL	Download-URL	MediaType	Format
Description)	GetCapabilities			
Web Accessible Folder (WAF)	https://woudc.org/home.php	https://woudc.org/archive/Archive-NewFormat/TotalOzone_1.0_1	text/html	-
Daten Suche / Download Benutzer Interface	https://woudc.org/data/explore.php?dataset=totalozone	-	text/html	-
Statischer Archiv Datensatz	https://woudc.org/home.php	https://woudc.org/archive/Summaries/datasetsnapshots/totalozone.zip	application/zip	-
OpenSearch Such Interface	https://api.eumetsat.int/data/search-products/os	-	-	application/opensearchdescription+xml

5.1.1.2 Schema.org

Im Gegensatz zu vielen OpenData Portalen sollen im umwelt.info Portal auch andere Ressourcentypen indexiert werden als die im vorhergehenden Kapitel vorgestellten. Hierfür eignet sich DCAT-AP(.de) nicht, da es ausschließlich für die Beschreibung von Datensätzen ausgelegt ist. Ein Vokabular mit wesentlich breiterem Anwendungsbereich ist schema.org. Das Ziel von schema.org ist es, generelle Schemata für strukturierte Daten im Internet und auf Webseiten bereitzustellen.

Schema.org bietet dafür derzeit ca. 800 unterschiedliche Typen zur Beschreibung von Informationen, darunter z. B. einfache Webseiten, Artikel, Videos und Mobile Apps. Aufgrund des Umfangs wird an dieser Stelle auf die Schemadefinitionen verwiesen [24]. Da viele Webseiten bereits mit schema.org annotiert sind, lassen sich die notwendigen Informationen bei einem Crawling von Websites relativ einfach in den Metadaten-Index des umwelt.info Portals übernehmen.

Es wird davon ausgegangen, dass die im umwelt.info Portal indexierten Ressourcentypen von dem generischen schema.org Typen „CreativeWork“ abgeleitet sind. Dieser definiert gemeinsame Eigenschaften der Ressourcen, wie URL, Beschreibung, Autor, Aktualisierungsdatum und räumlicher Ausschnitt. Zwar unterstützt schema.org auch einen speziellen Typen für die Beschreibung von Datensätzen. Dieser enthält allerdings nicht alle Eigenschaften aus DCAT-AP.de. Darum ist schema.org allein nicht hinreichend für umwelt.info. DCAT-AP.de als deutscher Standard für Open Data muss in vollem Umfang unterstützt werden.

Der Editor von umwelt.info sollte die Erfassung der folgenden Ressourcentypen unterstützen:

- Article
- MobileApplication
- WebApplication
- WebPage
- VideoObject
- Dataset

Zur Beschreibung dieser Ressourcentypen sollte die Erfassung der folgenden Eigenschaften unterstützt werden, beziehungsweise Elemente suchbar sein. Die hier vorgesehenen suchbaren Elemente und die Elemente, die für den MD-Editor vorgesehen sind, können bei Bedarf angepasst werden. Diese Anpassungen können zum Beispiel im Rahmen der Umsetzungsphase erfolgen.

Tabelle 5: Schema für Metadaten gem. schema.org (nach [24])

Elementname (englisch)	Beschreibung	Datatype	Element in schema.org	MD-Editor
ID	Die ID ist repräsentiert eine eindeutige Kennung der Datenstruktur (z. B. URI, ISBN).	Text	schema:identifier	**x (obligatorisch, automatisch erzeugt)**
Description	Diese Eigenschaft enthält eine Freitextbeschreibung.	Text	schema:description	**x (obligatorisch)**
Title	Titel der Ressource.	Text	schema:name	**x (obligatorisch)**
Contact	Wer ist der Ansprechpartner für die Ressource.	Organization or Person	schema:contactPoint	**x**
Publisher	Wer publiziert die Ressource.	Organization or Person	schema:publisher	
Author	Person oder Organisation, die die Daten erstellt hat.	Organization or Person	schema:author	
Theme	Kategorie (Theme) der Ressource. Hier lassen sich über einen entsprechenden Thesaurus vorgegebene Themen beschreiben.	Thing	schema:about	**x**

Elementname (englisch)	Beschreibung	Datatype	Element in schema.org	MD-Editor
CreationDate	Datum der Herausgabe bzw. Veröffentlichung der Ressource.	Date or DateTime	schema:dateCreated	**x**
ModificationDate	Letzten Änderung der Ressource.	Date or DateTime	schema:dateModified	**x**
Keyword	Schlagwort zur Ressource.	DefinedTerm or Text or URL	schema:keywords	**x**
Spatial Extent	Räumliche Ausdehnung, auf die sich die Informationsquelle bezieht.	Place	schema:spatialCoverage	**x**
Temporal Extent	Zeitliche Ausdehnung, auf die sich die Informationsquelle bezieht.	DateTime or Text or URL	schema:temportalCoveryge	**x**
License (s. 5.1.1.3)	Lizenz, unter welcher die Ressource zur Verfügung gestellt wird.	Creative Work or License	schema:license	**x**
URL	Link zur Ressource, z.B Homepage, Download.	URL	schema:url	**x (obligatorisch)**
Media-Type	Diese Eigenschaft bezieht sich auf das Format der Ressource und ist i.d.R. ein Medientyp der gemäß des von IANA definierten und zur Verfügung gestellten offiziellen Medientypregisters. Beispiele könnten IANA Media Typen sein für PDF, WMS, WCS, Shapefile.	URL	schema:encodingFormat	**x**

5.1.1.3 Lizenzen

Die Lizenz, unter welcher eine Ressource zur Verfügung gestellt wird, sollte idealerweise bereits in den Metadaten beschrieben sein. Sowohl DCAT-AP.de als auch schema.org bieten hierfür Felder an.

Allerdings kann der Inhalt dieser Felder nur so gut sein, wie die Information, die aus den Quell(meta-)daten abgeleitet wurden. Idealerweise sollte ein Link auf einen Lizenztyp (wie z. B. http://creativecommons.org/licenses/by-nc/3.0/) enthalten sein, um eine Lizenz eindeutig identifizieren zu können.

Minimal sollten aber vorhandene Beschreibungen von Lizenzen, Zugriffs- und Nutzungsrechten in den Ausgangs-Metadaten auf die entsprechenden „Lizenz-Attribute“ von DCAT-AP.de bzw. schema.org "ge-mappt“ werden. Die Adaptoren sind dann allerdings nicht in der Lage, die genauen Lizenzbedingungen abzuleiten. Dieses müsste dann händisch oder durch eine Qualitätssicherungs-Komponente für Lizenzen erfolgen. Einige Informationen zur Wiederverwendbarkeit werden bereits durch die vorgesehene MQA-Komponente (vgl. 2.1.8) überprüft.

5.1.1.4 Metadatenmodell der Suchmaschine

Datensätze und die anderen Ressourcentypen werden jeweils mit den Metadatenmodellen DCAT-AP.de und schema.org beschrieben. Da alle Ressourcen aber in einer gemeinsamen Suchmaschine indexiert werden sollen, muss es ein gemeinsames Schema für die Suche geben, in dem alle erforderlichen Ressourcentypen indexiert werden können.

Daneben müssen interne Statusinformationen des umwelt.info Portals abgelegt sein, die nicht direkt in den Metadaten enthalten sind. Für einen geharvesteten Metadatensatz ist das z. B. der Quellkatalog der Ressource.

Das Schema der Suchmaschine muss daher ein harmonisiertes Modell sein, das die grundlegenden Informationen beinhaltet, die für die Suche und die Ergebnisdarstellung (vgl. umwelt.info Designkonzept) aller indexierten Metadaten relevant sind.

Tabelle 6: Metadatenmodell für den Suchindex

Element-name (englisch)	Beschreibung	Datatype	Verbind-lichkeit	Suchbar
ID	Dieses Feld enthält die Haupt-ID der Datenstruktur.	Text	1	**x**
Document	Dieses Feld enthält die kompletten Metadaten im Ursprungsmodell (DCAT-AP.de oder schema.org).	Text	1	
Source Catalogue	Quellkatalog des Metadatensatzes, falls über Harvesting erstellt.	Text	0..1	**x**
Owner	E-Mail-Adresse des umwelt.info Nutzers, der für diesen Metadatensatz verantwortlich ist, falls über die manuelle Datenbereitstellung im Portal erfasst.	Text	0...1	**x**
Status	Status des Metadatensatzes, falls über die manuelle Datenbereitstellung im Portal erfasst.	draft, private, public	1	**x**

Element-name (englisch)	Beschreibung	Datatype	Verbind-lichkeit	Suchbar
Datestamp	Zeitpunkt des letzten Harvestings bzw. der letzten Editierung (vgl. UI: Benachrichtigungen)	DateTime	1	**x**
Validated	Stammt die Ressource von einer amtlichen Stelle. (vgl. UI: Facette „Datenbereitstellende“	Boolean	1	**x**
Title	Titel der Ressource.	Text	1	**x**
Description	Dieses Feld enthält eine Freitextbeschreibung.	Text	1	**x**
Contact	Wer ist der Ansprechpartner für die Ressource. „Contact“ wird ggf. auch über eine Zuordnungsliste der Quellen und Datenbereitsteller gepflegt. Für Datenquellen gem. dcat-ap.de kann initial die „contributors”-Liste [24] des Standards eingetragen werden.	Text	1	**x**
Theme	Kategorie (Theme) der Ressource. Hier lassen sich über einen entsprechenden Thesaurus vorgegebene Themen beschreiben.	Text	0..*	**x**
Modification Date	Letzte Änderung der Ressource (vgl. UI: Sortierung nach Aktualisierung)	DateTime	0..1	**x**
Spatial Extent	Dieses Feld bezieht sich auf die räumliche Abdeckung, welche durch die Datenstruktur abgedeckt wird.	Geometry	0..1	**x**
Location	Dieses Feld bezieht sich auf eine geographische Region, welche durch die Datenstruktur abgedeckt wird.	Text	0..1	**x**
Temporal Extent Begin	Dieses Feld bezieht sich auf den Beginn der zeitlichen Dauer, welche durch die Datenstruktur abgedeckt wird.	DateTime	0..1	**x**
Temporal Extent End	Dieses Feld bezieht sich auf das Ende der zeitlichen Dauer, welche durch die Datenstruktur abgedeckt wird.	DateTime	0..1	**x**
Format	Diese Eigenschaft bezieht sich auf das Format der Ressource (vgl. UI: Facette „Formate“) und ist i.d.R. ein Medientyp der gemäß des von IANA definierten und zur Verfügung gestellten offiziellen Medientypregisters.	Text	0..*	**x**

Element-name (englisch)	Beschreibung	Datatype	Verbind-lichkeit	Suchbar
	Beispiele könnten IANA Media Typen sein für PDF, WMS, WCS, Shapefile.			
URL	Link zur Ressource.	Text	1..*	
Resourcetype	Art der Ressource (vgl. UI: Facette „Inhalt"	Article, MobileApplication, WebApplication, WebPage, VideoObject, Dataset	1	**x**
DetailViews	Anzahl der Aufrufe der Detailseite zum Metadatensatz (siehe UI: Sortierung nach Anzahl Aufrufe)	Integer	1	
Keywords	Liste von Schlagwörtern (vgl. UI: Suchergebnisseite, Detailseite)	Freie Liste von Texten	0..*	**x**
License	Angabe der Lizenz (vgl. UI: Suchergebnisseite, Detailseite)	Freie Liste von Texten	0..*	
API	Angabe des Standards oder Typs der API, falls eine API für den Abruf der Daten verfügbar ist. (vgl. UI: Suchergebnisseite, Detailseite)	Freie Liste von Texten	0..*	
Properties	Aus anderen Textfeldern abgeleitete Eigenschaftenliste (vgl. UI: Facette „Eigenschaften")	Maschinenlesbar, API, Kostenlos, Datensatz, Freie Lizenz, Amtliche Datenbereitstellende	0..*	**x**

Für die Umsetzung des Suchschemas sind Mappings der beiden Metadatenmodelle zum internen JSON-basierten Schema der Suchmaschine erforderlich.

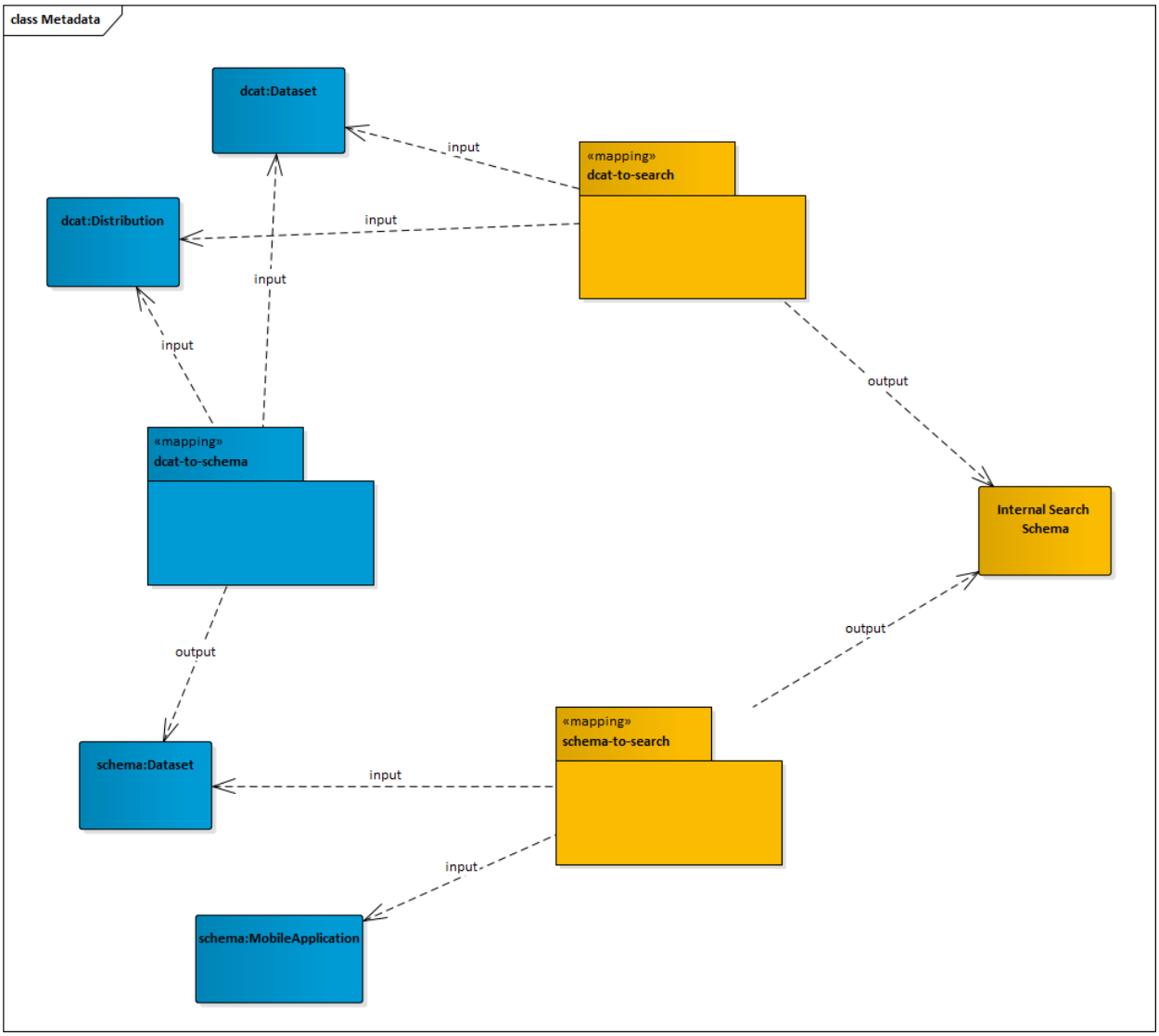
Einige Felder sind dabei speziell für die UI erforderlich. Es wird für die Benutzeroberfläche kein eigenes Feld für Kategorien benötigt, aber ein Mapping der Kategorien kann den Redakteur*innen helfen, Suchen für die Kacheln/Erkunden-Seitentypen zu definieren. Im Designkonzept sind darüber hinaus Eigenschaften an den Suchergebnissen und Detailansichten symbolisiert, die im Metadaten-Index im Feld „Properties" abgelegt sind. Ob für einen Datensatz die Eigenschaft „Maschinenlesbar" eingetragen sein muss, kann aus dem Datentyp abgeleitet werden (z. B. csv,

wms, etc.). Das Feld „Validated“, bzw. die Eigenschaft „Amtliche Datenbereitstellende“ leiten sich aus den Datenbereitstellenden, bzw. den geharvesteten oder gecrawlten Quellen ab. Dazu muss eine Liste gepflegt werden, die von den Harvestern und Crawlern genutzt wird. Eine erste initiale Befüllung dieser Zuordnungsliste kann auf der contributors Liste von dcat-ap.de [24] basieren. Ein analoges Vorgehen kann auch für das Ableiten von „Freie Lizenz“ und „Kostenlos“ erfolgen. Die Felder API und Lizenz, sowie deren Eigenschaften müssen hier aus den Informationen zum Format bzw. der Nutzungs- und Lizenzbedingungen abgeleitet werden.

Wird beim Harvesten oder Crawlen des Metadatensatzes kein Spatial Extent gefunden, wird versucht, über die Location auf einen Spatial Extent zu „mappen“. Dazu muss eine Zuordnungsliste gepflegt werden, die die Locations für den MD-Editor einem Spatial Extent zuordnet.

Es muss außerdem ein Mapping der Metadaten Elemente des DCAT-AP.de Modells nach schema.org geben. Dies ergibt sich aus der Anforderung, dass über eine “Representational State Tranfer” (REST) Schnittstelle eine mit schema.org annotierte HTML-Seite für jeden Metadatensatz generiert werden muss (siehe 2.10.2), also auch für Datensätze, die mit DCAT-AP.de beschrieben sind. Abbildung 31 zeigt die erforderlichen Mappings mit jeweils zwei exemplarischen Typen aus schema.org und DCAT-AP.de.

Abbildung 31: Erforderliche Mappings zwischen DCAT-AP.de, schema.org und dem internen Schema der Suchmaschine. Der Übersichtlichkeit halber sind nicht alle Typen aus DCAT-AP.de und schema.org dargestellt.



Quelle: eigene Darstellung, con terra GmbH

Für ein Mapping zwischen DCAT-AP und schema.org gibt es bereits Vorarbeiten der EU [25]. Diese sind auch für das Mapping der Metadaten im umwelt.info Portal zu nutzen. Wichtig für das Mapping ist, dass nicht nur das Schema, sondern auch die Inhalte der Metadaten berücksichtigt werden. Beispielsweise ist für die Kategorisierung von DCAT-AP Datensätzen das kontrollierte Vokabular der MDR-Themen zu verwenden. Allerdings ist dabei noch nicht vollständig geklärt, wie diese in schema.org übernommen werden kann. Im umwelt.info Portal sollten jedenfalls sowohl DCAT-AP.de als auch schema.org Metadaten in Kategorien klassifiziert werden können.

5.1.2 Benutzerinformationen

5.1.2.1 Benutzerprofil

Das Benutzerprofil enthält alle Informationen, die bei der Registrierung im umwelt.info Portal erfasst werden. Die dargestellte Tabelle enthält nur die Elemente, die aus fachlicher Sicht notwendig sind. Das Profil kann bei Bedarf um weitere Elemente erweitert werden (z. B. für Passworthistorie etc.)

Tabelle 7: Datenmodell für das Benutzerprofil

Element-name	Beschreibung	Datatype	Verbindlichkeitsstufe
Passwort	Anmeldepasswort. Das Passwort muss eine festgelegte Stärke besitzen und über ein geeignetes Passwort-Hashing-Verfahren gespeichert werden	String	1
E-Mail	E-Mail-Adresse des Nutzers	String	1
Name	Vollständiger Name des Nutzers	String	1
Institution	Institution, bei der der Nutzer beschäftigt ist	String	1
Anschrift	Straße und Hausnummer der Institution	String	0..1
Stadt	PLZ und Stadt	String	0..
Rolle	Zugeteilte Rollen im umwelt.info Portal. Ein Nutzer kann mehrere Rollen besitzen. Die bekannten Rollen aus den Use Cases sind: • Administrator • CMS-Redakteur • Metadaten-Redakteur • Registrierter Nutzer	Enumeration	0..*

5.1.2.2 Personalisierte Inhalte

Personalisierte Inhalte werden für alle registrierten Nutzer*innen des umwelt.info Portals gepflegt. Die personalisierten Inhalte sind jeweils nur für den betreffenden Nutzer sichtbar.

Die im Folgenden beschriebenen Entitäten sollen dabei nur verdeutlichen, welche Informationen generell erforderlich sind. Die konkrete technische Implementierung kann auch davon abweichen, solange die gewünschte Funktionalität gewährleistet ist.

5.1.2.2.1 Benachrichtigungen

Benachrichtigungen werden nicht direkt durch den Nutzer gepflegt, sondern automatisch vom umwelt.info Portal generiert. Benachrichtigungen werden erzeugt für personalisierte Inhalte, Favoriten und Gemerkte suchen (siehe unten).

Tabelle 8: Datenmodell für Benachrichtigungen

Element-name	Beschreibung	Datatype	Verbindlichkeitsstufe
Nutzername	Der Anmeldename des Nutzers, dem dieses Objekt zugeordnet ist	String	1
ID	Eindeutige ID dieser Benachrichtigung	String	1
Typ	Typ der Benachrichtigung	String	1
Status	Gelesen / nicht gelesen	Boolean	1
Zeitstempel	Datum und Uhrzeit der Erstellung	DateTime	1
Details	Freie Struktur mit Detailinformationen, z. B. für: • Aktualisierter Datensatz: Datensatz-ID • Neue Suchergebnisse: Suchergebnis-ID	Any	1

Einmal täglich soll das System den Status der Metadatensätze in den Favoriten überprüfen und ggf. über Aktualisierungen benachrichtigen. Der Status wird auf „gelesen“ gesetzt, wenn die Benachrichtigung in der UI quittiert wurde (z. B. mittels Anklickens eines Benachrichtigungssymbols).

5.1.2.2.2 Favoriten

Ein Favorit bezieht sich auf einen Metadatensatz aus dem Metadaten-Index des Portals. Die Favoriten werden durch die Nutzer*innen selbst gepflegt und sind in Listen organisiert. Wenn ein Favorit aktualisiert wird, wird durch das umwelt.info Portal eine Benachrichtigung für den betreffenden Metadatensatz erzeugt.

Tabelle 9: Datenmodell für Favoriten

Element-name	Beschreibung	Datatype	Verbindlichkeitsstufe
Nutzername	Der Anmeldename des Nutzers, dem dieses Objekt zugeordnet ist	String	1
Listenname	Name der Favoritenliste, zu der dieser Eintrag gehört	String	1

Element-name	Beschreibung	Datatype	Verbindlichkeitsstufe
Listenposition	Position in der Favoritenliste	Integer	1
ID	Eindeutige ID dieses Favoriten	String	1
Zeitstempel	Datum und Uhrzeit der Erstellung	DateTime	1
Aktualisierung	Datum und Uhrzeit der letzten Aktualisierung des Metadatensatzes	DateTime	1
Metadaten-ID	Verweis auf den Metadatensatz über eindeutige ID	String	1

5.1.2.2.3 Gemerkte Suchen

Registrierte Nutzer*innen des umwelt.info Portals können ihre Suchen speichern, um diese zu einem späteren Zeitpunkt erneut abzurufen. Die Gemerkten Suchen werden durch die Nutzer*innen selbst gepflegt. Wenn sich die Trefferanzahl für eine gespeicherte Suche gegenüber der letzten Ausführung erhöht, wird eine Benachrichtigung vom umwelt.info Portal generiert.

Tabelle 10: Datenmodell für gemerkte Suchen

Element-name	Beschreibung	Datatype	Verbindlichkeitsstufe
Nutzername	Der Anmeldename des Nutzers, dem dieses Objekt zugeordnet ist	String	1
ID	Eindeutige ID dieser Suche	String	1
Titel	Titel für diese Suche, der in dem Portal angezeigt wird	String	1
Zeitstempel	Datum und Uhrzeit der Erstellung	DateTime	1
Aktualisierung	Letzte Ausführung der Gemerkten Suche	DateTime	1
Treffer	Trefferanzahl bei der letzten Ausführung	Integer	1
Suchdefinition	Der Filter zur Ausführung, sowie weitere Einstellungen wie Sortierreihenfolge	Any	1

5.1.3 Datenaustauschformat

Bei dem Austausch von Daten sollte sich an den existierenden Standardformaten (z. B. GeoJSON, JSON, YAML, XML, CSV, TIFF, JPEG, usw.) der Community orientiert werden. Neu entwickelte Standardformate (z. B. XÖV) haben es schwer, sich auf dem Markt zu etablieren. Diese bestehen

meist aus komplexen Strukturen und sind daher für die Endanwender*innen nicht nutzbar. Auch die Verwendung neuer Datenformate in Softwareprodukten wird nicht empfohlen, da diese mit großer Wahrscheinlichkeit nicht unterstützt werden. Die Erfahrungen aus dem Open Data Bereich zeigt, dass es sinnvoll ist, gut beschriebene Daten zu veröffentlichen, welche sich nach Möglichkeit an den Open Data Prinzipien richten.

Des Weiteren wird auch durch den ITZ-Bund vorgegeben, um welche Datenformate es sich handeln muss. Diese sind in den Technischen Randbedingungen in der Tabelle 1 im Umsetzungskonzept als „Strukturierten Daten zum Zweck des Datenaustausches“ aufgeführt.

5.2 Architektur- und Entwurfsmuster

5.2.1 Starke Kohäsion und schwache Kopplung

Bei der Zerlegung in Komponenten sind ein starker Zusammenhalt (Kohäsion) und die lose Kopplung von Komponenten die maßgeblichen Ziele. Dabei führen eine lose Kopplung und ein starker Zusammenhalt der Komponenten zu einer wart- und testbaren Architektur. Einzelne Komponenten sind so ebenfalls besser wartbar und können unabhängig von anderen Komponenten getestet und deployed werden.

5.2.2 Microservices

Das Konzept der Microservices[6] unterstützt die lose Kopplung und den starken Zusammenhalt von Komponenten. Microservices stellen unabhängige Funktionen des Systems dar, die jeweils eine bestimmte Teilaufgabe erledigen und miteinander über sprachunabhängige Programmierschnittstellen kommunizieren. Darüber hinaus sind Microservices stark spezialisiert und unabhängig deploy- und testbar, indem sie jeweils eine REST-Schnittstelle bereitstellen, über die andere Microservices mit ihr kommunizieren.

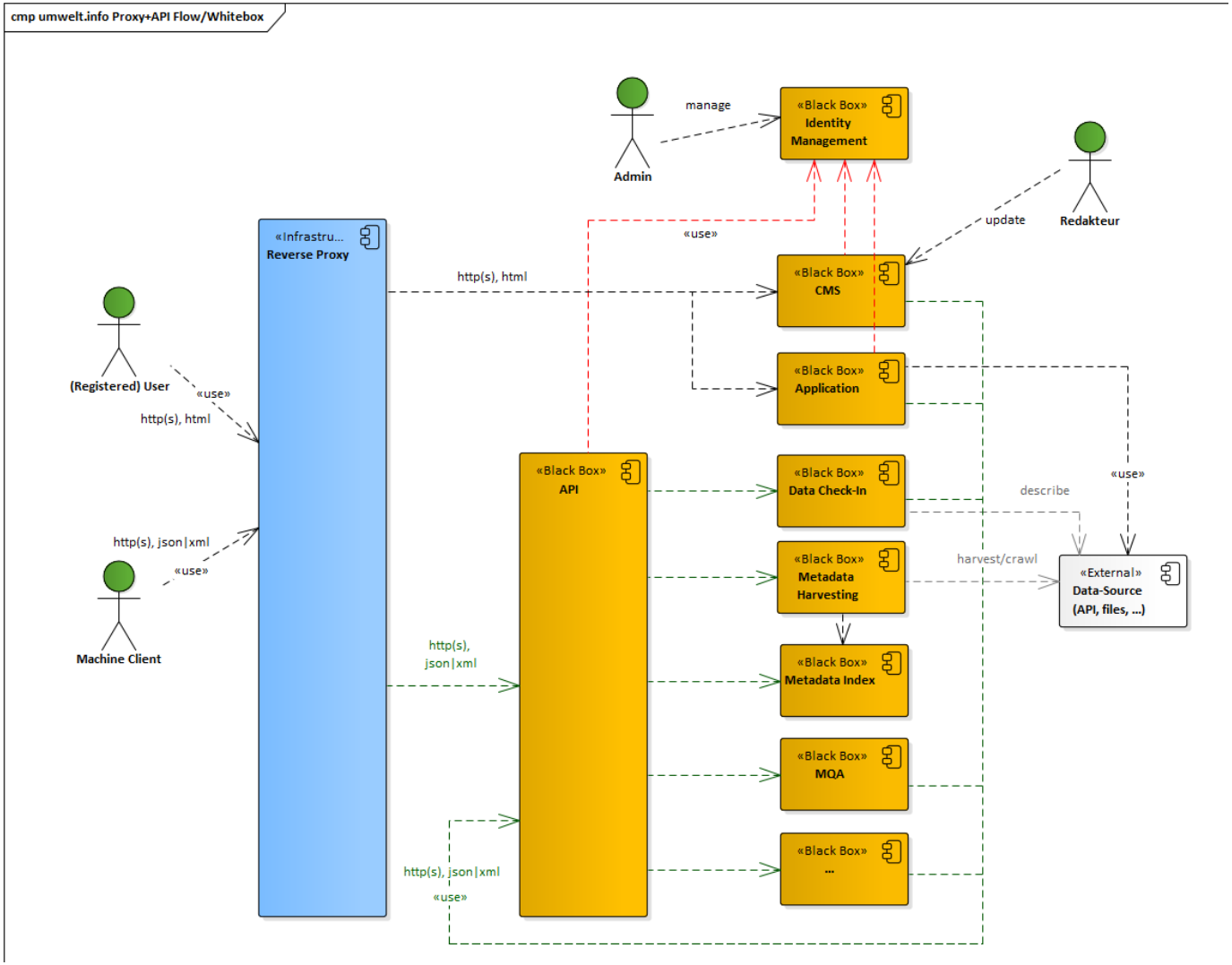
5.2.3 Reverse Proxy und API-Gateway

Der Zugriff auf die Komponenten des Portals von außen geschieht einheitlich über einen “Uniform Resource Locator” (URL) und wird durch eine Proxy-Komponente gesteuert, die auf die einzelnen Komponenten, in Abhängigkeit der angefragten URL weiterleitet bzw. „rückwärts“ überschreibt. Darüber hinaus werden API-Anfragen, die bestimmte APIs der einzelnen Komponenten betreffen über das API-Gateway der API-Komponente geleitet. Abbildung 32 zeigt die Kommunikationsbeziehungen der Komponenten über http(s), unter Einsatz der Reverse Proxy-Infrastrukturkomponente und der API-Komponente.

Im Vergleich zu den in Abbildung 1 dargestellten, logischen Abhängigkeiten der Komponenten untereinander sind die Kommunikationswege optimiert. Die Komplexität wird erheblich reduziert und die Wartbarkeit der Infrastruktur somit erhöht. Zum Beispiel lässt sich eine Komponente oder ein Server austauschen, ohne dass dies in allen abhängigen Komponenten umkonfiguriert werden muss, da nur der Proxy den eigentlichen Ort kennen muss. Zudem wird nach außen nur eine einzige Adresse benötigt und die internen Adressen der Komponenten und Server bleibt nach außen verborgen.

[6] https://www.gartner.com/en/information-technology/glossary/microservice

Abbildung 32: Reverse Proxy und API



Quelle: eigene Darstellung, con terra GmbH

5.3 Entwicklungskonzepte

5.3.1 DevOps

DevOps[7] ist ein modernes Konzept, das die Belange von Entwicklung (Development) und Betrieb (Operation) vereinheitlicht (DevOps). Dazu richtet der DevOps-Ansatz den Blick auf die Ausrichtung und den Einsatz von entsprechenden Teams, Prozessen und Werkzeugen. Diese sollen so eingesetzt und aufeinander abgestimmt werden, dass (neu) entwickelte oder verbesserte Komponenten schnell und mit hoher Qualität deployed werden können.

5.3.2 Wart- und Testbarkeit

Um eine optimale Wart- und Testbarkeit zu erreichen, soll das Gesamtsystem in kleine Einheiten aufgeteilt werden, die unabhängig voneinander entwickelt, geändert, getestet, deployed und skaliert werden können. Die Kapitel 3.3 (Lösungsstrategie) und 2 (Bausteinsicht) in diesem Dokument, folgen diesem Prinzip bereits. Durch die Realisierung unter Einsatz von Microservices werden zudem die Prinzipien des starken Zusammenhaltes und der losen Kopplung unterstützt. Die einzelnen Microservices lassen sich unabhängig warten und testen. Unit Tests auf Komponentenbasis ermöglichen dabei das automatische Testen mit geringem Aufwand. Alle Unit Tests werden dann nach der Änderung einer Komponente erneut ausgeführt.

5.3.3 Automatisiertes Delivery / Deployment (CI/CD)

Um vereinfacht und automatisiert neu entwickelte oder geänderte Funktionen bereitstellen zu können, sollen „Continuous Integration“[8]-Pipelines (für die Test- und Entwicklungsumgebungen), sowie „Continuous Delivery“-Pipelines (für das Produktivsystem) verwendet werden. Durch den Einsatz von CI/CD-Pipelines können verkürzte Test- und Updatezyklen, im Sinne des DevOps, erreicht werden. Weitere Vorteile sind automatisierte Tests (vgl. 5.3.2, Unit Testing), Reports und ein einheitlicher Zugriff auf das System.

5.4 Betriebskonzepte

5.4.1 DevOps

Siehe 5.3.1.

5.4.2 Cloud-Infrastruktur

Eine „Public Cloud“ ist eine Infrastruktur, die von einem Dienstleister bereitgestellt wird. Ihr Vorteil ist, dass kein Kapital in Rechner- und Datenzentrumsinfrastruktur investiert werden muss. Eine „Private Cloud“ wird eigens für eine bestimmte Organisation betrieben. Das Hosten und Verwalten einer „Private Cloud“-Plattform kann intern oder durch externe Dienstleister erfolgen. Dabei kann die Infrastruktur im eigenen oder in einem Rechenzentrum eines Dienstleisters betrieben werden. Ein hybrides Modell bietet einen kombinierten Zugang zu eigenen und externen Infrastrukturen. Diese können dann nach den Bedürfnissen ihrer Nutzer*innen gestaltet werden. Ein hybrides

[7] https://www.gartner.com/en/information-technology/glossary/devops

8 https://www.gartner.com/en/information-technology/glossary/continuous-integration-ci

Modell kann zum Beispiel bei Lastspitzen oder Systemänderungen zum Einsatz kommen, wenn die eigenen Ressourcen in bestimmten Situationen nicht ausreichen.

5.4.3 Containerisierung

Ein moderner Ansatz für den Betrieb von Systemkomponenten in Cloud-Infrastrukturen ist Containerisierung. Dazu werden Container betrieben, die durch ein Container-Image definiert sind. Dabei ist ein Container eine Instanz eines Images, die alle für seine Aufgabe erforderlichen Abhängigkeiten (z. B. Dateien, Softwarepakete, Frameworks, Bibliotheken, etc.) enthält. Dadurch sind Container über ihre Images in jeder Phase (Entwicklung, Test, Betrieb, etc.), plattformunabhängig, konsistent und portierbar. Sie besitzen Eigenschaften, die für die Entwicklung einer Microservice-basierten Architektur ideal sind:

- Stand Alone: Alle erforderlichen Abhängigkeiten sind in dem Container enthalten. Er stellt dadurch die komplette Laufzeitumgebung in einem Paket bereit. Dadurch wird er portabel und ist unabhängig vom Host, der die Container ausführt.
- Isolation: Ein Container besitzt nur sehr eingeschränkte Zugriffsrechte auf das Betriebssystem und die Prozesse des Hosts.

Für diese Betriebsform werden dann nur noch Container-Images entwickelt, die in der Cloud-Infrastruktur eingesetzt werden. Dieser Ansatz ermöglicht zum Beispiel das flexible Updaten oder Austauschen eines Containers oder das Betreiben mehrerer Instanzen einer Komponente. Damit werden durch den Einsatz von Cloud-Produkten und Containerisierung die benötigte Skalierungs- und CI/CD-Konzepte unterstützt.

Viele Cloud-Anbieter unterstützen den Einsatz von containerisierten Softwarekomponenten als Dienstleistung [26], [18]. Bei dieser Bereitstellungsform der Infrastruktur entfällt die Pflege und Wartung eines eigenen Clusters auf eigenen (virtuellen) Maschinen.

5.4.4 Skalierbarkeit

Um zu gewährleisten, dass sich bei einer Zunahme der Nutzungszahlen des umwelt.info Portals die Antwortzeiten bei Suchanfragen oder bei der Benutzung von Elementen der Benutzerschnittstelle nicht merklich verschlechtern sowie die Zeit bis zur Aktualisierung der Metadaten bei einer Zunahme an zu harvestenden Datenquellen nicht merklich zunimmt, muss das System skalierbar sein. Das bedeutet in diesem Fall, dass die Hard- und Software Ressourcen des Systems sich automatisch an die gestiegenen Anforderungen anpassen. Hierfür eignen sich besonders Cloud-basierte Systeme. „Public Cloud“ Systeme sind ideal geeignet, wenn ad-hoc IT-Ressourcen hinzugefügt werden sollen und wenn große Skalierungsmöglichkeiten benötigt werden. „Private Clouds“ reichen bezüglich ihrer Skalierungsfähigkeiten oft aus und bieten häufig mehr Sicherheit und Datenschutz.

5.4.5 Logging, Monitoring, Tracing

Für Administrator*innen ist es für den Betrieb des umwelt.info Portals wichtig, hochwertige Daten über den Zustand des Systems zur Verfügung zu haben, um Ausfälle und Fehler zu finden und zu beheben. Dies wird durch die moderne, verteilte Microservice-Architektur und die hohe Parallelisierung von Abläufen erschwert, da bei Auftreten eines Problems alle beteiligten Komponenten gefunden und deren Zustand erfasst werden müssen. Alle Komponenten müssen deswegen aussagekräftige Logausgaben mit konfigurierbarer Granularität erzeugen. Die Logs der

einzelnen Komponenten können dann mit Log-Aggregatoren gesammelt werden, um einen einheitlicheren Blick auf den Zustand des Gesamtsystems zu erhalten.

Ebenso sollten Metriken über die Systemkomponenten verfügbar und analysierbar sein. Dafür werden kontinuierlich Kennzahlen über die verwendeten Systemressourcen, z. B. Random Access Memory (RAM), CPU, und Verfügbarkeit der Komponenten gesammelt. Diese dienen dazu, die Systemressourcen zu optimieren oder Aussagen über ein potenzielles Nutzungsszenario zu machen. Sie können außerdem als Eingabedaten für ein Monitoringsystem verwendet und über Dashboards dargestellt werden.

Oft ist es für die Lösung eines Problems wichtig, eine einzelne Anfrage durch alle beteiligten Systemkomponenten zu verfolgen. Distributed Tracing ist eine Methode, dies zu ermöglichen. Hierfür gibt es bereits einzelne Frameworks, die dann in den Code integriert werden müssen. Bestehende Cloud-Infrastrukturen bieten hierfür teilweise auch schon automatisierte Lösungen an und erleichtern die Implementierung erheblich.

5.5 Sicherheitskonzepte

In diesem Kapitel werden wesentliche und wichtige Inhalte zur Sicherheit der Architektur beschrieben, welche bei der Umsetzung berücksichtigt werden sollten.

Im umwelt.info Portal kommt es durch die Registrierung am CMS zur Verwendung von personenbezogenen Daten. Für die Eingabe der personenbezogenen Daten wird ein zentralisiertes Identity Management System zum Einsatz kommen. Bei der Vergabe von Passwörtern sollten diese sich an vorgegebenen Passwortrichtlinien orientieren und durch eine Mindestanforderung bzgl. Komplexität verfügen. Außerdem sollte nach der Registrierung im Identity Management System eine Verifizierung der Nutzer*innen durch Bestätigung über E-Mail möglich sein. Bei Verlust des Passworts sollte es den Nutzer*innen möglich sein, dieses zurückzusetzen und ein neues zu erstellen. Kommt es bei dem Login zu schnell aufeinander folgenden gescheiterten Versuchen, sollte eine temporäre Kontosperrung erfolgen, um diese vor unbefugtem Zugang und Angriffen auf das System zu schützen.

Alle Portalfunktionen müssen über einen dedizierten Berechtigungsschutz verfügen. Administrative Funktionen wie z. B. die Nutzerverwaltung dürfen nur den dafür vorgesehenen Personenkreis zugänglich sein.

Beim Austausch von sensiblen Daten sollten empfohlene kryptographische Verfahren vom „Bundesamt für Sicherheit in der Informationstechnik“ (Symmetrische Verschlüsselung, Asymmetrische Verschlüsselung, Kryptographische Hashfunktionen, Datenauthentisierung, Instanzauthentisierung, Schlüsselvereinbarung, Secret Sharing, Zufallszahlengeneratoren) verwendet werden, um vor unbefugtem Zugriff zu schützen und dadurch den Austausch zu sichern [27].

Bei der technischen Umsetzung des umwelt.info Portals werden Konfigurationen von Systemeigenschaften, Datenbanksystemen benötigt sowie Quellcode generiert. Damit diese Konfigurationen und der Quellcode (z. B. in Git) nicht verloren gehen, sollten diese über ein regelmäßiges Backup gesichert werden. Dadurch ist es möglich, bei Ausfall auf einen definierten Systemzustand zurückzugreifen, um das System zeitnah wiederherzustellen zu können.

Um die Einhaltung der Vorgaben und Anforderungen des umwelt.info Portal sicherzustellen, wird ein Auditor empfohlen. Dabei kann es sich um mehrere Personen handeln, welche Zugriff auf die Kontrollmechanismen des Systems besitzen und die Einhaltung der Vorgaben und Anforderungen

überprüfen. Hierzu werden vor allem die Prozesse zur Metadatensuche und Registrierung am Portal, sowie die Stabilität des Systems geprüft. Dadurch sollen Schwachstellen und Qualitätsverbesserungen gefunden werden. Über Logging und Monitoring Mechanismen, können Schwachstellen ausfindig gemacht werden (siehe Kapitel 5.4.5).

Damit das umwelt.info Portal vor Angriffen und unbefugten Zugriffen geschützt ist, sollten nur für das System relevante Softwarepakte auf den Hosts des umwelt.info Portals installiert werden. Dabei wird vor allem der Einsatz von dedizierter Software empfohlen, welche das System nach außen stärkt. Dabei sollte nur Software zum Einsatz kommen, welche für den Betrieb notwendig ist. Die Systemkomponenten müssen sich auch mit eingeschränkten Nutzerberechtigungen ausgeführt werden können (kein Ausführen als „root“).

Für die Kommunikation des umwelt.info Portals mit externen Schnittstellen sollte auf den neuesten Standards der Verschlüsselung aufgesetzt werden. Diese Verschlüsselung basiert auf TLS/SSL Protokollen, welche den Datenaustausch bei der Übertragung sicherstellen.

Eingaben von den Nutzenden des Portals müssen vom System validiert werden. Dies betrifft zum einen Eingaben, mit denen auf die internen Datenbanken des Systems (z. B. Nutzerdatenbank, Metadaten-Index) zugegriffen werden kann. Die Eingaben dürfen nicht ungeprüft weitergegeben werden, um Angriffe wie z. B. SQL Injection zu verhindern. Andererseits soll die Qualität der Metadaten im umwelt.info Portal durch eine obligatorische Validierung sichergestellt werden. Dafür sollte vor der Bereitstellung von neuen Metadaten durch die Nutzer*innen im umwelt.info System eine Validierung der Metadaten erfolgen. Dadurch wird sichergestellt, dass es sich um qualitativ hochwertige Informationen handelt und keine unpassenden Inhalte im umwelt.info Portal vorgehalten werden. Bei der Eingabe von Metadaten in der Metadatenmaske sollte stehts eine Validierung erfolgen, welche auf Vollständigkeit prüft und jedes Eingabefeld durch die Nutzer*innen ausgefüllt wurden ist.

Für die Umsetzung des umwelt.info Systems wird sehr stark empfohlen, sich an den „Technischen Richtlinien“ [28] des „Bundesamt für Sicherheit in der Informationstechnik“ [29] zu orientieren.

5.6 Künstliche Intelligenz

In diesem Kapitel liegt das Hauptaugenmerk auf Techniken, mit denen KI-Funktionen im Suchkontext realisiert werden können. Dabei spielen für die Architektur einerseits moderne KI-Methoden eine Rolle, aber auch die gewünschten Funktionen und wie diese effizient betrieben werden können.

Bei der Entwicklung von „suchnahen“, Index-anfragenden Komponenten können spezielle Techniken eingesetzt werden, die dem Bereich der Künstlichen Intelligenz zugeordnet werden. Diese suchnahen Komponenten werden von der API und dem CMS genutzt und sind dem Metadaten-Index „vorgeschaltet“: KI-Suche, RS, Search RS, Search Prediction, und QA Retrieval (vgl. Abbildung 3: Ebene 1 umwelt.info: KI und Linked Data Optionen (White Box)).

Dieses Kapitel soll Einstiegspunkte für die Entwicklung der KI-Komponenten liefern. Daher sind die hier beschriebenen Konzepte auf Suchtechnologie fokussiert. Entwickler und Data Scientists können darüber hinaus weitere Techniken bei der Verarbeitung von Textdaten einsetzen.

Ein querschnittliches Konzept für KI-Funktionen sind spezielle Listen (etwa mittels NLP-Methoden ermittelte Begriffe), die von Suchfunktionen effizient abgefragt werden können. Semantische Anfragen an den Metadatenindex können bei der Realisierung von KI-Funktionen helfen. Bei der Verarbeitung der Suchanfrage können beliebige NLP-Techniken eingesetzt werden. Schließlich sollten auch weitere Datengrundlagen (außerdem Metadatenindex selbst) berücksichtigt werden.

5.6.1 Sammlungen von Texten für KI-Funktionen

Für einige Funktionen ist der kanonische Ansatz eine (kuratierte) Liste von Begriffen oder Texten (ggf. in einem eigenen Suchindex), aus der dem Nutzende ausgewählte Einträge im Rahmen einer bestimmten Funktion präsentiert werden können. Eine solche Liste kann etwa für die Search Prediction oder Search-RS-Komponente vorgehalten werden. Zum Beispiel funktioniert eine einfache Implementierung der Search RS-Komponente auf Basis einer Vorschlagsliste, die mit einer entsprechenden Suche abgefragt wird (vgl.2.2.5).

Für die Erstellung einer solchen Suchliste können Daten aus dem Index analysiert werden (etwa mittels der in diesem Kapitel dargestellten Methoden) oder andere Datengrundlagen herangezogen werden (etwa vorhandene Thesauren oder weitere Textdatenbanken).

Ein Vorteil der Verwendung von Listen ist die aus der Indexierung resultierenden Skalierbarkeit und Performanz der so realisierten Funktionen. Ein weiterer Vorteil ist ein hohes Maß an Kontrolle, welchen die Redakteure über die darüber realisierten KI-Funktionen erhalten (es werden keine Texte angezeigt, die nicht gelistet sind). Ein Nachteil dieses Ansatzes ist ein erhöhter Wartungsaufwand für die Pflege von rein manuell angelegten Listen.

Es können auch frühere Sucheingaben gesammelt und ausgewertet werden, um Listen zu erstellen und zu verbessern (vgl. 5.6.4). Außerdem können auch zusätzliche Datenquellen ausgewertet werden, um weitere Funktionen zu ermöglichen. So kann zum Beispiel Twitter genutzt werden, um Vorschläge zu aktuellen Themen zu pflegen.

Der Einsatz von beliebigen weiteren NLP-Methoden ist bei der Erstellung und Pflege von Listen denkbar.

5.6.2 Semantische Indexabfragen

Um semantisch „verwandte“ Begriffe zu ermitteln, können für einen Suchbegriff alle mit diesem Begriff im Index zusammen auftretenden Begriffe abgefragt und ausgewertet werden. Dazu werden zwei Strukturen des Suchindexes benötigt.

- Der „Invertierte Index“ speichert für alle Begriffe, die in einem Textfeld vorkommen, eine Liste der Metadatensätze, in denen der Begriff vorkommt.
- Der „Forward Index“ speichert für alle Metadatensätze die enthaltenen Begriffe.

Über den Invertierten Index lassen sich alle Metadatensätze abfragen, die einen Begriff enthalten. Über den Forward Index lassen sich alle in den gefundenen Metadatensätzen vorkommenden Begriffe abfragen. In Solr lassen sich über diese beiden Strukturen „co-occurence“ oder „relatedness“ für Begriffe ermitteln. Dabei ist „relatedness“ eine Funktion die Solr bereitstellt und die bei Indexabfragen genutzt werden kann (vgl. „Structure of a semantic knowledge graph“ in [12]).

Semantisch verwandte Begriffe können zum Beispiel genutzt werden, um Suchvorschläge zu machen, Intents zu erkennen oder um Suchanfragen zu „expandieren“. Dabei können verwandte Begriffe ggf. mit geringer Gewichtung in die Suche eingeschlossen werden.

Begriffe können auch mit vordefinierten Thesauren, Taxonomien, Ontologien oder Linked Data abgeglichen werden, um darüber Suchanfragen oder Suchvorschläge besser auf umwelt.info „anzupassen“. Im Gegensatz zur rein inhaltsbasierten Analyse ist darauf zu achten, dass so

gefundene, verwandte Begriffe zwar sprachlich sinnvoll sind, aber ohne ein vorhandenes Feld im Index nicht sichergestellt ist, dass diese Begriffe auch zu Suchtreffern führen.

5.6.3 Sprachmodelle und Worteinbettung

Ein moderner Ansatz aus dem Bereich der NLP-Methoden setzt auf die Repräsentation von Textfragmenten (etwa Sucheingaben, Textfelder im Index, etc.) als „Worteinbettungen“ (auch „Embeddings“). Embeddings sind Vektorrepräsentationen von Textfragmenten, die semantische Zusammenhänge und den Kontext der Wörter im Textfragment als Vektoren repräsentieren sollen. So soll neben dem jeweiligen Wort auch der jeweilige (Satz-)Kontext erfasst werden.

Um Embeddings zu erzeugen, muss ein fein abgestimmtes (Transformer-Encoder-)Modell trainiert werden. Dabei wird häufig ein vortrainiertes, generelles Modell herangezogen (etwa ein auf den gesamten Wikipedia Datenbestand trainiertes) und mit einem domänen- und aufgabenspezifischen Textkorpus „nachtrainiert“.

Für den Einsatz eines solchen Sprachmodells für eine Suchfunktion sind die folgenden Schritte bei der Indexierung und bei der Abfrage nötig (siehe „Applying Transformers to Search“ in [12]; vgl. 2.11).

Folgende Schritte werden für die Indexierung durchgeführt:

- Extraktion von „concepts“ aus dem Index-Korpus.
- Embeddings für die ausgewählten „concepts“ generieren („Dense Vector Representation“).
- Embeddings in einem Index speichern.

Folgende Schritte werden für die Suche im Index durchgeführt:

- Embeddings für Sucheingabe generieren („Dense Vector Representation“).
- Suche im Index, etwa mittels „Approximate Nearest Neighbour“.
- Gegebenenfalls Threshold für Similarity Score anwenden.

Embeddings für Textfragmente sind vielseitig nutzbar. So können außer einer Verbesserung der Suchergebnisse auch Question-Answering-Systeme (vgl. 2.2.7) realisiert werden oder Suchvorschläge als „Natural Language Autocomplete“ umgesetzt werden [12].

Der Nachteil dieses Konzeptes ist der erhöhte Rechenaufwand bei der Nutzung von großen Sprachmodellen.

5.6.4 Nutzungstracking als Datenbasis für KI-Funktionen

Ein weiterer Ansatz für die Verbesserung von Index-Anfragenden Komponenten (Suche, Autovervollständigung, etc.) basiert auf der Analyse von getrackten Nutzungsdaten. Dabei können die gleichen Techniken zum Einsatz kommen wie bei der Analyse von Inhalten im Index. Für eine Autovervollständigung kann etwa eine Sammlung von (erfolgreichen) Sucheingaben indexiert werden oder mittels semantischer Abfrage ausgewertet werden.

Der Vorteil von Suchanfragen als Datenbasis liegt darin, dass diese Daten von den Suchenden selbst stammen. So können zum Beispiel Suchbegriffe vorgeschlagen werden, die Metadatensätze betreffen, die nur wenige textuelle Beschreibungen beinhalten. Im Gegensatz dazu würden

ausschließlich inhaltsbasierte Suchvorschläge solche Metadatensätze naturgemäß weniger berücksichtigen. Somit eignet sich die Nutzung von Trackingdaten besonders in Kombination mit einem gepflegten Index für Suchvorschläge (vgl. 5.6.1). Aus Datenschutzgründen sollte keine vollautomatische Verwendung von früheren Sucheingaben der Nutzenden zum Einsatz kommen.

Ein weiterer Einsatzzweck von Trackingdaten fällt in den Bereich der automatischen Verbesserung des Rankings für Suchergebnisse. Hierzu können zunächst Metadatensätze mit einem „Boostingfaktor“ für bestimmte Anfragen belegt werden, um etwa häufig angeklickte Metadatensätze für eine bestimmte Suchanfrage höher zu ranken (vgl. „Popularized Relevance through Signals Boosting“ in [12]). So kann mit einem einfachen Boosting-Ansatz ein bestimmter Metadatensatz für spezielle Anfragen als besonders wichtig markiert werden.
Dieses Verfahren lässt sich mittels KI-Ansatz automatisieren. Dazu kann ein genereller, sogenannter „Learning to Rank“-Algorithmus trainiert werden. Für ein LTR-basiertes Ranking werden Trainingsdaten benötigt, die wie folgt strukturiert sind:

- Suche (etwa „Schweinswal Ostsee“),
- Metadatensatz (ID im Index),
- Features (etwa Numerische Felder, der ermittelten Relevanz für die Suche auf Textfeldern, weitere semantische Abfragen, Berechnungen, etc.).
- Judgement (relevant / nicht relevant)

Mittels der Judgements werden Metadatensätze als relevant oder nicht relevant für eine Suche gekennzeichnet. Die Features werden als Abfragen formuliert und die Trainingsdaten auf dem Index abgefragt. So lassen sich die Trainingsdaten für das LTR-Modell generieren. Ein LTR-Modell wird offline trainiert und dann wiederum im Index, für das Ranking in Suchanfragen genutzt. Die Feature-Definitionen werden zusätzlich als Abfragen definiert und im Index gespeichert. So können die Features von einem gespeicherten LTR-Modell genutzt werden. Das Verfahren ist auch unter „Learning to rank for generalizable search relevance“ in [12] beschrieben.

Abschließend lässt sich der Boosting- und LTR-Ansatz auch auf Trackingdaten, statt manueller Relevanz-Judgements anwenden. Hierbei werden die Judgements aus den Nutzungsdaten ermittelt (siehe „Building learning to rank training data from user clicks“ in [12]).

5.7 Spatial Data on the Web (SDW) / Linked Data (LD) (E9)

5.7.1 Linked Data-Prinzipien

Die Prinzipien des Web und für Linked Data besagen:

- alle Metadaten (Metadaten Ressourcen) sind identifiziert durch persistente URLs
- alle Metadaten sind im HTML-Format über ihre URL abrufbar, so dass sie von Menschen einfach mittels Web-Browser gelesen werden können. Hierdurch sind die Metadaten zudem auch von etablierten Internet-Suchmaschinen aufzufinden und indizierbar.
- alle Metadaten sind in einem RDF-basierten [30] strukturierten Format (z. B. RDF/XML, JSON-LD) vorliegend und über ihre URL abrufbar, so dass sie von (Such-)Maschinen gelesen und verstanden werden können. Auch von Web-Entwicklern sind diese Formate nutzbar.

- Alle mittels HTML repräsentierten Metadaten werden durch Vokabulare klassifiziert, die auch von den Internet-Suchmaschinen des Web unterstützt werden, d. h. sie sind mit schema.org oder DCAT [21] annotiert und können z. B. damit etwa auch über „Google Dataset Search“ gefunden werden.
- die Interaktion mit den Metadaten basiert auf dem HTTP-Protokoll und ist konsistent mit dessen Design (Unterstützung von HTTP-Verben, content negotiation, etc.)
- APIs zum Zugriff auf die Metadaten sollen selbsterklärend sein (z. B. als OpenAPI Service-Beschreibung vorliegen)
- Herstellung und Verwaltung von Links zwischen Metadaten. Solche Links können dynamisch zugeordnet werden

Auf der Basis dieser Prinzipien lassen sich verschiedene Anwendungen entwickeln. In umwelt.info werden so insbesondere ein SPARQL-Client (siehe Kap. 4.9 im Umsetzungskonzept) und Mashups als erweiterte Antworten zu einer Sucheingabe (siehe Kap. 4.6 im Umsetzungskonzept) ermöglicht. Auch lassen sich die verlinkten Informationen zur Unterstützung verschiedener Suchverfahren verwenden (s. z. B. Kap 5.6.2).

5.7.2 RDF und „Linking“

Die Grundlage für das Repräsentieren der Metadaten in einer Linked Data konformen Art ist das Ressource Description Framework (RDF) [30]. Hier sind Metadaten als Tripel, bestehend aus Subjekt, Prädikat, Objekt repräsentiert. Dabei ist der Metadatensatz (der einen Datensatz beschreibt) die eigentliche Ressource (Subjekt), die über eine URL referenzierbar ist. Die einzelnen Eigenschaften des Metadatensatzes sind jeweils durch ein Prädikat repräsentiert, welches auf ein Objekt (den Eigenschaftswert) verweist. Die Prädikate sind ebenfalls über eine URL definiert, worüber sich weitere Informationen zum Prädikat erfragen lassen. Die Objekte können einfache Texte oder Zahlen sein aber eben auch andere, „verlinkte“ Ressourcen (z. B. eine Datensatz-Serie (Collection), ein Sensor, eine Adresse, ein geographisches „Feature“ oder ein Thema). Über die URL der Ressource „Metadatensatz“ lassen sich die Informationen in menschenlesbaren (z. B. HTML) oder in maschinenlesbaren (z. B. XML oder JSON-LD [31]).

5.7.2.1 Herstellung von Links

Neben den geplanten use cases (z. B. Daten-Stories) bei denen manuell links zwischen Metadaten von Datensätzen hergestellt werden, ist es sinnvoll, wenn links automatisch zwischen Ressourcen hergestellt werden können. Hierzu gibt es bereits eine Reihe von maschinellen Ansätzen zum automatisierten Ver-Linken von RDF-Daten (z. B. Tools wie SILK [32]). Automatisches Linken kann etwa über die semantische Nähe verwendeter Begriffe geschehen. Ein weiterer Ansatz ist etwa das Linken über den geographischen Bezug (z. B. LIMES [33]), den Zeitpunkt/Zeitraum und/oder über die Begriffe eines gemeinsam verwendeten Thesaurus (z. B. der UMTHES). Da die Ressourcen in der vorliegenden Domäne auf DCAT-AP beruhen, vielfach einen Raum- und Zeitbezug besitzen und häufig Begriffe aus gleichen Thesauren verwenden, besteht eine hohe Chance automatisch Links zwischen ihnen erzeugen zu können.

6 Entwurfsentscheidungen

Dieses Kapitel reflektiert wichtige und riskante Architekturentscheidungen, für die eine Auswahl für eine oder mehrere Alternativen getroffen werden kann und für die eine lokale Beschreibung (z. B. in der White Box-Sicht von Bausteinen in Kapitel 2) nicht erfolgt ist. Andere grundlegende Entscheidungen, die sich zum Beispiel direkt aus den Fachlichen Anforderungen ergeben, sind bereits in Kapitel 3.3 des Umsetzungskonzepts beschrieben.

Die folgenden Unterkapitel geben Empfehlungen in Form eines Architecture Decision Record (ADR), um wichtige Entwurfsentscheidungen strukturiert festzuhalten. Aufgrund des Kriteriums „Aktualität“ werden Entscheidungen dabei vorrangig als Empfehlung genannt, um zum Zeitpunkt der Umsetzung eine Neubewertung zu ermöglichen.

Die hier beschriebenen Empfehlungen basieren auf den folgenden Kriterien:

- Vorgaben durch technische Randbedingungen (vgl. Kapitel 3.1 im Umsetzungskonzept)
- Aktualität und Stand der Technik
- Bewertung der Eignung eines Musters oder einer Komponente (z. B. auf Basis von Erfahrungswerten)
- Auswirkungen, Technische Schulden, Kosten, Folgekosten
- Entwicklungsstand des Produktes / Aktivität der Community / Pflege durch Hersteller
- Open Source oder proprietäre Software

6.1 Suchmaschine: Elasticsearch (ELK-Stack)

Beschreibung:

Die Suchmaschine Elasticsearch wird von Elasticsearch B.V. vertrieben.

Die oben genannten Kriterien werden weitestgehend erfüllt. Weitere Aspekte werden im Folgenden zum Vergleich aufgeführt.

Verteilung:

- Native Unterstützung für Clusterverteilung
- Deployment für Kubernets vom Hersteller vorbereitet

Index:

- Lucene (Benötigte Feldtypen werde unterstützt: Facetten, Textfelder, Datumsfelder, etc.)

NLP / Semantische Analysen:

- Benötigt eigene Entwicklung
- Vector Search wird unterstützt

Lizenz:

Es gibt Einschränkungen im proprietären Teilen des ELK-Stacks bezüglich Weitervertrieb

Weitere Softwareprodukte im Elasticsearch-Umfeld sind Logstash, das auf die „ingestion“ von Text- und insbesondere von Logdateien spezialisiert ist; sowie Kibana, das auf Datenauswertungen spezialisiert ist. Im Kern wird für umwelt.info die Suchmaschine benötigt, welche in Elasticsearch auf Lucene basiert. Für einige Komponenten können vorhandene Werkzeuge des ELK-Stacks hilfreich sein.

Status:

Offen

6.2 Suchmaschine: Solr

Beschreibung:

Die Suchmaschine Solr wird von der Apache Software Foundation entwickelt.

Die oben genannten Kriterien werden weitestgehend erfüllt. Weitere Aspekte werden im Folgenden zum Vergleich aufgeführt.

Verteilung:

- Native Unterstützung für Clusterverteilung

Index:

- Lucene (Benötigte Feldtypen werde unterstützt: Facetten, Textfelder, Datumsfelder, etc.)

NLP / Semantische Analysen:

- Benötigt eigene Entwicklung
- Vector Search wird unterstützt

Lizenz:

Apache Software License 2.0.

Solr wurde ursprünglich für den Use Case „Textsuche“ entwickelt. Die vorhandenen Funktionen und Aktivitäten der Community sind entsprechend fokussiert. Diese Ausrichtung kann für umwelt.info hilfreich sein. So sind etwa die in diesem Dokument beschriebenen KI-Funktionen angelehnt an semantischen Analysen, für die es Beispiele in Solr gibt.

Status:

Offen

6.3 Cloudinfrastruktur statt Betrieb eigener Hardware

Beschreibung:

- Verschiedene Anbieter stellen fertige Lösungen für Kubernetes-basierte, skalierbare Cloudinfrastrukturen bereit.
- Einfache dynamische Skalierbarkeit der Ressourcen bei hoher Auslastung des Systems.

- Hohe Flexibilität beim Bedarf der Ressourcen, welche durch den Anbieter bereitgestellt werden.
- Reduzierung der Betriebs- und Wartungsarbeiten von IT-Ressourcen.
- Keine Investitionskosten bei der Anschaffung von Hardwareressourcen.
- Steigerung der Datensicherheit durch die Nutzung des Cloud-Computing.
- Schnelle Integration von zusätzlichen Komponenten in das System.

Status:

Empfehlung

6.4 Kubernetes

Beschreibung:

Es wird eine Umgebung für Containerisierte Systemkomponenten benötigt. Kubernetes ist der Defacto-Standard für die Orchestrierung von containerisierten Anwendungen. Mit Kubernetes als Clusterinfrastruktur sind die Softwarekomponenten von der Infrastruktur entkoppelt und einzelne Komponenten skalierbar.

Status:

Empfehlung

6.5 Eigene Entwicklung von Harvester bzw. Crawler-Workflows mit Deskriptoren

Beschreibung:

Ein eigener Lösungsansatz für die Orchestrierung der Verarbeitungsschritte der Harvester und Crawler wurde in Kapitel 2.5, Ebene 2 umwelt.info – Metadata Harvesting und Crawling (White Box), beschrieben.

Vorteile:

- Genau zugeschnittene Lösung und optimale Integration mit den anderen Komponenten
- Freie Wahl bei Lizenzierung und Dienstleistern

Nachteile:

- Aufwand für Pflege und Erweiterungen

Status:

Option

6.6 Einsatz eines Standardproduktes als Workflow-Engine für Harvester bzw. Crawler-Workflows

Beschreibung:

Eine alternative Lösung für den Betrieb von Container-Workflows ist die Nutzung einer Standardkomponente. Dabei weicht die Nutzung einer Workflow-Engine von der beschriebenen Architektur ab (vgl. Kapitel 2.5, Ebene 2 umwelt.info – Metadata Harvesting und Crawling (White Box)). Bekannte Produkte sind zum Beispiel argo [32] oder Apache Airflow [33].

Vorteile:

- Verbreitetes Produkt für den Einsatz in containerisierten Umgebungen
- Aktive Community
- Regelmäßige Updates
- Open Source Lösung
- Flexible Erweiterung und Entwicklung von Workflowdefinitionen

Nachteile:

- Umfangreiche Software und Komplexe Lösung für vergleichsweise einfache Workflows

Die Architektur von Argo unterscheidet sich von Apache Airflow insoweit, dass argo Workflows mit Kubernetes-Mitteln realisiert, d. h. Workflows sind selbst Kubernetes-Artefakte.

Status:

Option

6.7 Vue.js als Frontend-Framework

Beschreibung:

Durch den Einsatz von Vue.js können unabhängige Front-End-Komponenten entwickelt und Bibliotheken mit fertigen, modernen Front-End-Komponenten genutzt werden (z. B. Vuetify [32]. Das Framework wird aktiv von einer Community und den Entwicklern gepflegt und wird frei unter der MIT-Lizenz vertrieben.

Kernkonzepte und Merkmale von Vue.js sind u.a. [34]:

- Virtual DOM
- Model-View-Viewmodel
- Deklaratives Rendering
- Komposition von Komponenten
- Serverseitiges Rendering

Im Vergleich zu anderen Frontend-Frameworks kann Vue.js einfacher in bestehende Websites integriert werden. Vue.js wird allerdings noch nicht als Front-End-Framework in den Architekturrichtlinien des ITZ Bund geführt.

Status:

Empfehlung

6.8 DBMS für die Datenhaltung

Beschreibung:

Verschiedene Komponenten können unterschiedliche DBMS nutzen. Vorgaben hierbei sind nur für eigens entwickelte Komponenten sinnvoll.

Einige Vorgaben sind in den technischen Randbedingungen genannt. Es handelt sich dabei um folgende mögliche DBMS:

- MariaDB
- Microsoft SQL Server
- Oracle Database
- Oracle MySQL
- PostgreSQL

Bei den genannten DBMS handelt es sich bis auf die Oracle Database, um lizenzfreie Produkte. Die Verwendung der einzelnen DBMS ist von Fall zu Fall entscheiden.

Status:

Offen (Fallabhängig)

6.9 Webserver

Beschreibung:

Beide Webserver werden zur Übertragung von Inhalten zwischen Client und Server genutzt. Dabei ist die Verwendung einer der beiden Webserver von Fall zu Fall entscheiden.

Der AppacheHTTP ist ein etabliertes langjähriges weitverbreitets Produkt auf den Markt und überzeugt mit seiner einfachen Konfiguration. Der nginx wurde gegenüber dem AppacheHTTP für hohe Last und Geschwindigkeit entwickelt. Der große Unterschied besteht darin, dass der AppacheHTTP bei jeder Anfrage an den Client einen Prozess startet und dadurch eine hohe Auslastung auf dem Server entstehen kann. Der nginx verarbeitet die Anfragen an den Client über Ausführung mehrerer Prozesse, welche in einem einzigen Prozess laufen und dadurch die Anzahl großer Anfragen effizienter verarbeiten kann. Es handelt sich bei beiden Webservern, um Open Source Softwareprodukte [35].

Status:

Offen (Fallabhängig)

7 Quellenverzeichnis

[1] Dr. Gernot Starke, Dr. Peter Uruschka und Mitwirkende, „arc42, das Template zur Dokumentation von Software- und Systemarchitekturen.," [Online]. [Zugriff am 19.08.2021].

[2] G. Börner, M. Bluhm, T. Fechner, R. Illes, B. Lubahn, M. Ostkamp, S. Richter, M. Schromm, U. Voges und J. Van Zadelhoff, „Umwelt- und Naturschutzinformationssystem UNIS-D - Machbarkeitsstudie," [Online]. Available: https://www.umweltbundesamt.de/publikationen/umwelt-naturschutzinformationssystem-unis-d. [Zugriff am 19.08.2021].

[3] Umweltbundesamt, „Semantischer Netzwerk Service (SNS)," [Online]. Available: https://sns.uba.de/de. [Zugriff am 14.09.2021].

[4] Open Geospatial Consortium, „OpenGIS Catalogue Services Specification 2.0.2 - ISO Metadata Application Profile: Corrigendum," 15 02 2017. [Online]. Available: https://portal.ogc.org/files/?artifact_id=77855. [Zugriff am 01 09 2021].

[5] Initial Operating Capability Task Force for Network Services, „Technical Guidance for the implementation of INSPIRE Discovery Services," 07 11 2011. [Online]. Available: https://inspire.ec.europa.eu/file/1551/download?token=Ws4F8_aS. [Zugriff am 01 09 2021].

[6] SmartBear Software, „OpenAPI Specification Version 3.0.3," 20 02 2020. [Online]. Available: https://swagger.io/specification/. [Zugriff am 08 07 2021].

[7] „Metadata Quality Assessment Methodologie des European Data Portals," [Online]. Available: https://data.europa.eu/mqa/methodology?locale=de. [Zugriff am 22.09.2021].

[8] „FAIR Guiding Principles for scientific data management and stewardship," [Online]. Available: https://www.go-fair.org/fair-principles/. [Zugriff am 22.09.2021].

[9] „Data Quality Vocabular, W3C," [Online]. Available: https://www.w3.org/TR/vocab-dqv/. [Zugriff am 22.09.2021].

[10] „matomo," [Online]. Available: https://matomo.org/matomo-vs-google-analytics-comparison/. [Zugriff am 14.10.2021].

[11] T. Grainger und T. Potter, Solr in Action, Manning Publications, 2014, p. 664.

[12] T. Grainger, D. Turnbull und M. Irwin, AI-Powered Search, Manning Publications, 2022 (voraussichtlich).

[13] F. Kirstein, K. Stefanidis, B. Dittwald, S. Dutkowski, S. Urbanek und M. Hauswirth , „Piveau: A Large-Scale Open Data Management Platform Based on Semantic Web Technologies," in ESWC, Cham, 2020.

[14] Elastic Search, „https://www.elastic.co/de/elasticsearch/features," [Online]. Available: https://www.elastic.co/de/elasticsearch/features. [Zugriff am 31 03 2022].

[15] Apache Solr, „https://solr.apache.org/features.html," Apache Solr, [Online]. Available: https://solr.apache.org/features.html. [Zugriff am 31 03 2022].

[16] D. Turnbull und J. Berryman, Relevant Search, Manning Publications, 2016.

[17] Rasa. [Online]. Available: https://rasa.com/blog/rasa-x-community-edition-changes/. [Zugriff am 13 06 2022].

[18] „Docker Container in Amazon aws," [Online]. Available: https://aws.amazon.com/de/containers/ . [Zugriff am 01.09.2021].

[19] „Kubernetes," [Online]. Available: https://kubernetes.io/de/docs/home/ . [Zugriff am 07.2021].

[20] „DCAT," [Online]. Available: https://www.w3.org/TR/vocab-dcat/. [Zugriff am 07.2021].

[21] „dcat-ap.de," [Online]. Available: https://www.dcat-ap.de/. [Zugriff am 07.2021].

[22] „DCAT-AP-JRC," [Online]. Available: https://ec-jrc.github.io/dcat-ap-jrc/. [Zugriff am 05.10.2021].

[23] „schema.org," [Online]. Available: https://schema.org/docs/schemas.html. [Zugriff am 07.2021].

[24] „Amtliche Datenbereitstellende / contributors, dcat-ap.de," [Online]. Available: https://www.dcat-ap.de/def/contributors/ . [Zugriff am 08.09.2021].

[25] „dcat-ap to schema.org," [Online]. Available: https://ec-jrc.github.io/dcat-ap-to-schema-org. [Zugriff am 07.2021].

[26] „Docker container in Microsoft azure," [Online]. Available: https://azure.microsoft.com/de-de/product-categories/containers/ . [Zugriff am 01.09.2021].

[27] „Schlüssellängen, BSI-Technische Richtlinie zu Kryptographische Verfahren: Empfehlungen und," [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile. [Zugriff am 13.10.2021].

[28] BSI, „Technische Richtlinien des BSI," [Online]. Available: https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/technische-richtlinien_node.htm. [Zugriff am 04.10.2021].

[29] BSI, „BSI," [Online]. Available: https://www.bsi.bund.de/DE/Home/home_node.html. [Zugriff am 04.10.2021].

[30] „RDF," [Online]. Available: https://www.w3.org/RDF/. [Zugriff am 07.2021].

[31] BSI, „Strukturierte Daten für Datensätze," BSI, [Online]. Available: ttps://developers.google.com/search/docs/appearance/structured-data/dataset?hl=de. [Zugriff am 13 10 2021].

[32] A. C. B. J. V. P. P. R. Isele, „SILK," University of Mannheim, [Online]. Available: http://silkframework.org/. [Zugriff am 17 11 2022].

[33] U. L. Institut für Angewandte Informatik, „LIMES," Institut für Angewandte Informatik, Universität Leipzig, [Online]. Available: https://aksw.org/Projects/LIMES.html. [Zugriff am 17 11 2022].

[34] „argo workflows," [Online]. Available: https://argoproj.github.io/workflows. [Zugriff am 14.09.2021].

[35] „Apache Airflow," [Online]. Available: https://airflow.apache.org/. [Zugriff am 14.09.2021].

[36] https://vuetifyjs.com/en/, „Vuetify," [Online]. Available: https://vuetifyjs.com/en/. [Zugriff am 04.10.2021].

[37] „Nginx vs. Apache: Wann welcher Webserver sinnvoll ist," [Online]. Available: https://t3n.de/news/nginx-vs-apache-814684. [Zugriff am 4 Oktober 2021].