TEXTE
65/2025

Abschlussbericht

Data Cube – Daten zur Umwelt

umwelt.info, Teilvorhaben 2

von:

Dennis Wilhelm, Dr. Ramona Sasse, Marcel Sprotte, Thomas Everding
con terra GmbH, Münster

Dr. Heino Rudolf
hrd.consulting, Dresden

Herausgeber:
Umweltbundesamt

Für Mensch & Umwelt

Umwelt Bundesamt

TEXTE 65/2025

Ressortforschungsplan des Bundesministeriums für
Umwelt, Naturschutz und nukleare Sicherheit

Forschungskennzahl 3720 12 101 0

FB001475

Abschlussbericht

Data Cube – Daten zur Umwelt

umwelt.info, Teilvorhaben 2

von

Dennis Wilhelm, Dr. Ramona Sasse, Marcel Sprotte,
Thomas Everding
con terra GmbH, Münster

Dr. Heino Rudolf
hrd.consulting, Dresden

Im Auftrag des Umweltbundesamtes

Impressum

Herausgeber
Umweltbundesamt
Wörlitzer Platz 1
06844 Dessau-Roßlau
Tel: +49 340-2103-0
Fax: +49 340-2103-2285
buergerservice@uba.de
Internet: www.umweltbundesamt.de

Durchführung der Studie:
con terra GmbH
Martin-Luther-King-Weg 20
48155 Münster

Abschlussdatum:
Januar 2024

Redaktion:
Fachgebiet I 1.5 Nationale und internationale Umweltberichterstattung
Michel Frerk

Publikationen als pdf:
http://www.umweltbundesamt.de/publikationen

ISSN 1862-4804

Dessau-Roßlau, Mai 2025

Kurzbeschreibung: Data Cube – Daten zur Umwelt

Dieser Abschlussbericht beschreibt ein Projekt zur technischen Neugestaltung der „Daten zur Umwelt“, mit dem das Umweltbundesamt (UBA) umfangreiche Umweltinformationen im Internet zur Verfügung stellt. Das Projekt wird unter dem Titel „Data Cube“ geführt. In der aktuellen Form bietet das Online-Angebot vor allem statische Daten, welche nur mit einem hohen manuellen Aufwand zur Verfügung gestellt werden können. Ziel dieser Konzeption ist es, eine nachhaltige Lösung zu entwerfen, mit der Umweltdaten importiert, gespeichert und der Öffentlichkeit als interaktive Tabellen und Diagramme angeboten werden können.

Das Dokument ist in zwei wesentliche Teile gegliedert und spiegelt den zeitlichen Verlauf des Projektes wider. Zuerst wurde gemeinsam mit dem UBA eine Anforderungsanalyse durchgeführt, in welcher sowohl der allgemeine redaktionelle Prozess als auch die verschiedenen Komponenten Data-Input, Data-Store, Data-Explorer und Data-Output analysiert werden. Anschließend wurden verschiedene Lösungskomponenten ermittelt und gegen die Anforderungen abgeglichen. Basierend auf der Anforderungsanalyse und den dargestellten Lösungen, wird die .Stat Suite durch das UBA ausgewählt. Im Konzeptionsteil wird, basierend auf dieser Lösung, ein Ansatz zur Umsetzung der Daten zur Umwelt dargestellt.

Abstract: Data Cube – Data on the Environment

This final report describes a project for the technical redesign of the “Data on the Environment” (in German ”Daten zur Umwelt”) with which the German Federal Environment Agency (German abbreviation: UBA) provides comprehensive environmental information on the Internet. The project is known by the name “Date Cube”. In its current form, the online service offers mainly static data that can only be made available with a great deal of manual effort. The aim of the concept is to design a sustainable solution with which environmental data can be imported, stored and offered to the public as interactive tables and diagrams.

The document is divided into two main parts and reflects the chronological course of the project. First, a requirements analysis was carried out together with the UBA, in which both the general editorial process and the various components of Data-Input, Data-Store, Data-Explorer and Data-Output are analyzed. Subsequently, various libraries and tools were identified and compared against the requirements. Based on the requirements analysis and the solutions presented, the .Stat Suite is selected. In the conceptual part, an approach for implementing the data on the environment is presented based on this solution.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Abkürzung	Erklärung
.Stat DE	.Stat Data Explorer (Komponente der .Stat Suite)
.Stat DLM	.Stat Data Lifecycle Manager (Komponente der .Stat Suite)
AA	Allgemeine Anforderungen
API	Application Programming Interface
BI	Business Intelligence
BMU	Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit
BMWi	Bundesministeriums für Wirtschaft und Energie
CMS	Content Management System
CSV	Comma Separated Values, Abkürzung für ein Dateiformat für Tabellen
DE	Data-Explorer
DESTATIS	Statistisches Bundesamt, Wiesbaden
DI	Data-Input
DO	Data-Output
DS	Data-Store
DWD	Deutscher Wetterdienst
DzU	Daten zur Umwelt
ETL	Extract, Transform, Load
FB	Fachbereich
FG	Fachgebiet
FME	Feature Manipulation Engine, ETL Software von Safe Software
GML	Geography Markup Language
INSPIRE	Infrastructure for Spatial Information in the European Community (nach Richtlinie 2007/2/EG des Europäischen Parlaments und des Rates vom 14. März 2007)
ISO 191xx	Internationale Organisation für Normung; Serie zur Normung von Geoinformationen und Geodaten (Federal Geographic Data Committee, o. J.)
JDCB	Java Database Connectivity
JPG, JPEG	Joint Photographics Expert Group, Abkürzung für Dateiformat für Grafiken
KI	Künstliche Intelligenz

Abkürzung	Erklärung
OECD	Organisation for Economic Co-operation and Development (deutsch: Organisation für wirtschaftliche Zusammenarbeit und Entwicklung)
OKSTRA	Objektkatalog für Straßen- und Verkehrswesen
OLAP	Online Analytical Processing
PNG	Portable Network Graphics, Abkürzung für ein Dateiformat für Bilddateien
QS	Qualitätssicherung
REST	Representational State Transfer
SDMX	Statistical Data and Metadata eXchange
SIS-CC	Statistical Information System Collaboration Community
SQL	Structured Query Language
UBA	Umweltbundesamt
UCD	User-Centered Design
UML	Unified Modeling Language
umwelt.finfo	Umwelt- und Naturschutzinformationssystem - Deutschland
URL	Uniform Resource Locator
WFS	Web Feature Service
WMS	Web Mapping Service
XML	Extensible Markup Language
X-ÖV	XML in der öffentlichen Verwaltung
X-PLANUNG	XML für Bauleitpläne, Raumordnungspläne und Landschaftspläne

Zusammenfassung

Dieses Dokument beschreibt ein Projekt zur technischen Neugestaltung der Datenhaltung, Visualisierung und Bereitstellung der „Daten zur Umwelt“ (www.umweltbundesamt.de/daten/), über die das Umweltbundesamt umfangreiche Umweltinformationen im Internet zur Verfügung stellt. Das Projekt wird unter dem Titel „Data Cube“ geführt. In der aktuellen Form bietet das Online-Angebot bereits Daten aus verschiedensten Themenbereichen im Content-Management-System (CMS) des UBA. Die Inhalte sind vor allem statische Daten wie zum Beispiel Microsoft Excel-Tabellen oder Diagramme im PNG- oder JPG-Bildformat, die zum Download bereitstehen. Diese können nur mit einem hohen manuellen Aufwand zur Verfügung gestellt werden, da die Pflege zu großen Teilen händisch in Excel-Tabellen durchgeführt wird. Vereinzelt sind bereits interaktive Diagramme durch das Drupal-Modul Easychart umgesetzt. Allerdings bestehen weiterhin Probleme bei der Nutzung, da weder eine Datenselektion möglich ist noch die Daten maschinenlesbar ausgegeben werden können. Es bleiben einige manuelle Schritte notwendig, außerdem wird keine zentrale Datenhaltung als Quelle verwendet.

Ziel der Konzeption war es daher, eine nachhaltige und effiziente Lösung für den Data Cube zu entwerfen, durch die der gesamte redaktionelle Prozess zur Veröffentlichung von Umweltdaten vereinfacht und verbessert werden kann. Dieser Prozess beinhaltete den Datenimport (Data-Input), die zentrale Speicherung (Data-Store), die Organisation (Data-Explorer) und die Bereitstellung von interaktiven Tabellen und Diagrammen (Data-Output). Die Zielstellung wird eingehender in Kapitel 1 thematisiert.

Das Dokument ist im weiteren Verlauf in zwei wesentliche Teile gegliedert und spiegelt den zeitlichen Verlauf des Projektes wider. Initial wurde gemeinsam mit dem UBA eine Anforderungsanalyse (Kapitel 2) durchgeführt, in welcher die zuvor genannten Prozess-Teile untersucht wurden. Hierzu wurden zuerst zwei Workshops mit Teilnehmenden der Redaktion durchgeführt, um den aktuellen redaktionellen Prozess genauer zu verstehen. Die Analyse ergab zusammenfassend, dass ein hoher Kommunikationsaufwand der Redaktion mit den verschiedensten Interessengruppen existiert, welcher unter anderem durch den Austausch von Excel-Dateien geprägt ist. Gleichzeitig ist ein hohes Maß an Qualität gefordert, weshalb Daten und Texte mehrere Schritte vom Entwurf über Qualitätssicherung bis zur Freigabe benötigen. Zusätzlich zum redaktionellen Prozess wurden bestehende Datenportale anderer Organisationen gemeinsam analysiert. Basierend auf den Beispielen und den zusätzlichen Informationen der Teilnehmenden, wurden in dieser Phase ebenfalls die Anforderungen an die Data-Explorer und Data-Output Komponenten des Data Cubes formuliert. Zusätzlich zu den Workshops mit der Redaktion wurden thematische Gespräche mit einzelnen datenhaltenden Stellen geführt. Durch diese Gespräche wurde ein erster Überblick über die Heterogenität der Datenquellen gewährleistet, welcher als Basis für die Data-Input Konzeption notwendig ist. Weitere Gespräche mit datenhaltenden Stellen sollten während der Umsetzungsphase durchgeführt werden. Für die Komponente des Data-Stores wurde kein expliziter Workshop abgehalten. Die Anforderungen ergaben sich jedoch ebenso durch die gesichteten Quelldaten und den Anforderungen an die nutzenden Komponenten (Data-Explorer und Data-Output). Im weiteren Verlauf wurden alle Anforderungen in einer Anforderungsmatrix erfasst und nach den verschiedenen Komponenten kategorisiert.

Im zweiten Teil des Dokuments folgt die Konzeption, basierend auf der Anforderungsanalyse (Kapitel 3 und 4). Mit Hilfe einer reduzierten Anforderungsmatrix wurden zunächst 16 Programme/ Bibliotheken/ Tools vorausgewählt, welche für die Umsetzung des Data Cubes in Frage kommen könnten. Gemeinsam mit dem UBA wurde diese Auswahl auf die Lösungskomponenten .Stat Suite, Highcharts, Mesap, Sisense und Tableau reduziert. Für diese

Möglichkeiten wurde anschließend die umfassende Anforderungsmatrix ausgefüllt und jeweils ein Grobkonzept erarbeitet (Kapitel 3.5). Von den dargestellten Lösungskomponenten wurde die .Stat Suite durch das UBA zur finalen Konzeption ausgewählt, da diese nicht nur viele Anforderungen abdeckt, sondern insbesondere auch Open Source ist und aktiv weiterentwickelt wird. In Kapitel 4 des Dokuments wird ein Ansatz zur Umsetzung der Daten zur Umwelt, basierend auf der .Stat Suite, dargestellt.

Basierend auf der Konzeption ist in Kapitel 5 die Umsetzung des Projektes beschrieben. Dabei wird zunächst die IT-Infrastruktur auf Basis von Docker-Compose erläutert. Es folgt die Integration der Daten. Für die Verwendung der .Stat Suite wurden zuerst SDMX-Datenstrukturen erzeugt. Hierbei lag ein Fokus auf dem fachlichen Austausch mit den verschiedenen datenhaltenden Stellen, um ein möglichst einheitliches Datenmodell über den gesamten Data Cube zu etablieren. Es folgt die eigentliche Datenintegration in die .Stat Suite. Technisch wurde diese durch die Software FME gelöst. Es werden verschiedene Prozessschritte beschrieben, um Quelldaten für die .Stat Suite aufzubereiten. Weiterhin werden verschiedene Automatisierungen zum kontinuierlichen Aktualisieren von Datensätzen beschrieben.

Der Bericht endet mit der Beschreibung von weiteren Entwicklungsmöglichkeiten und einer abschließenden Einwertung des Vorhabens.

Summary

This document describes a concept for the technical redesign of the "Environmental Data", with which the German Federal Environment Agency (German abbreviation: UBA) provides comprehensive environmental information on the Internet. This project is known by the name “Data Cube”. In its current form, the online platform already provides data from a wide range of subject areas in the UBA Content-Management-System (CMS). The contents are mainly static data such as Microsoft Excel tables or diagrams in PNG- or JPG-format. These data are available as downloads. To make the data publicly available a structured process is carried out which currently involves many manual steps. It is documented and organized by using an Excel table. In some cases, interactive charts have already been implemented using the Drupal module Easychart. However, there are still problems with its use, as neither data selection is possible, nor can the data be output in machine-readable form. Thus, some manual steps remain necessary, and no centralised data storage is used as a source so far.

The aim of the concept is to design a sustainable solution that can simplify the entire editorial process for publishing environmental data. This process includes the data import (Data-Input), a central data storage (Data-Store), the organization of data (Data-Explorer) and the provision of interactive tables and charts (Data-Output). In chapter 1 the main target is described in more details.

The document is divided into two main parts and reflects the chronological course of the project. Initially, a requirements analysis (chapter 2) was carried out together with the UBA, in which the editorial process was examined. To this end, two workshops were first held with participants from the editorial team to gain a more precise understanding of the current process. In summary, the analysis showed that there is a lot of communication between the editorial team and various stakeholders required. Among other things, the exchange of Excel files is mentioned to be crucial for this. Moreover, the data and texts need to pass several stages from drafting to quality assurance to release to guarantee a high level of quality. In addition to the editorial process, existing data portals of other organizations were jointly analyzed. Based on the examples and the additional information provided by the participants, the requirements for the components Data-Explorer and Data-Output of the Data Cube were also formulated in this phase. Next to the workshops with the editorial team, thematic discussions were held with individual data-holding bodies. These discussions provided an initial overview of the heterogeneity of the data sources, which was crucial for the Data-Input concept. Further discussions with data-holding agencies were recommended during the implementation phase. No dedicated workshop was held for the Data-Store component. However, the requirements also resulted from the source data reviewed and the requirements for the using components (Data-Explorer and Data-Output). During the requirements analysis, all requirements were recorded in a table and categorized according to the different components.

The second part of the document contains the conceptual design based on the requirements analysis (chapters 3 and 4). With the help of a reduced requirements matrix, 16 programs, libraries and tools were pre-selected which qualified for the Date Cube implementation as they meet particular requirements. Together with the UBA, this selection was further condensed to the potential solution .Stat Suite, Highcharts, Mesap, Sisense and Tableau. The requirements matrix was then completed for these options and a rough concept was developed for each of those (chapter 3.5). Of the solution components presented, .Stat Suite was selected by the UBA for a final design concept because it not only covers many of the requirements, but is also open source and is actively being developed further. In chapter 4 of this report, an approach for implementing the environmental data based on the .Stat Suite is presented.

Chapter 5 describes the implementation of the project based on the concept. Firstly, the IT infrastructure based on Docker-Compose is explained. This is followed by the integration of the data. SDMX data structures were first created for the use of the .Stat Suite. The focus here is on the technical exchange with the various data-holding entities in order to establish a data model that is as standardized as possible across the entire data cube. This is followed by the actual data integration into the .Stat Suite. Technically, this was solved using the FME software. Various process steps are described in order to prepare source data for the .Stat Suite. Furthermore, various automations for the continuous updating of data sets are described.

The report ends with a description of further development possibilities and a final evaluation of the project.

1 Zielstellung und Gegenstand

1.1 Zielstellung

Eine Aufgabe des Umweltbundesamtes (UBA) ist die Umweltberichterstattung. Dies beinhaltet sowohl die Information der Öffentlichkeit über den Zustand der Umwelt in Deutschland als auch die politischen Berichtspflichten (Umweltbundesamt, 2020 - b). Ein wichtiges Werkzeug zur Erfüllung dieser Aufgabe ist das Angebot „Daten zur Umwelt“ (DzU), mit dem das Umweltbundesamt umfangreiche Umweltinformationen im Internet zur Verfügung stellt (Umweltbundesamt, o. J.). Im Bereich „Umweltzustand und Trends“ sind die Daten in 12 Themenbereiche untergliedert und werden in Form von Tabellen, Karten, Diagrammen und Infografiken für diverse Umweltthemen zur Verfügung gestellt. Auf diese Weise lassen sich eine Fülle an Informationen finden. Allerdings handelt es sich meist um statische Informationen wie zum Beispiel Grafiken als Bild oder PDF-Dokument oder um Exceldateien zum Download. Ein interaktives Arbeiten mit den Daten oder das Kombinieren von unterschiedlichen Daten wird derzeit nicht ermöglicht. Der Aufwand zur redaktionellen Aufbereitung der Daten für die Präsentation im Internet ist derzeit sehr hoch, da der Grad der Automatisierung gering ist (Umweltbundesamt, 2020 - a). Ziele sind daher im Wesentlichen [angepasst nach (Umweltbundesamt, 2020 - a)]:

- Reduzierung des Aufwands für das UBA: Dies gilt für den gesamten Arbeitsprozess von der Übernahme von Daten inklusive Qualitätssicherung bis zur visuellen Aufbereitung und Datenabgabe. Die Aufwandsreduzierung ist insbesondere im Hinblick auf immer größer werdende Datenmengen unumgänglich. Die weltweite Datenzunahme wird für den Zeitraum 2018 bis 2025 auf 530% geschätzt (Europäische Kommission, o. J. - c), was vermutlich für Umweltdaten ebenso anzunehmen ist.
- Flexibilisierung von Auswertungen: Es sollen Möglichkeiten geschaffen werden, Datenbestände zu erkunden, für den jeweiligen Nutzungszweck flexibel zu analysieren, nutzergruppengerecht aufzubereiten und in unterschiedlichen Datenformaten bereitzustellen.
- Erschließung neuer Technologien: Schaffung von Voraussetzungen für weitergehende und zukunftsweisende Auswertungen, wie zum Beispiel Künstliche Intelligenz, Virtual oder Augmented Reality Anwendungen.
- Verzahnung mit dem Umwelt- und Naturschutzinformationssystem Deutschland (umwelt.info): Die „Daten zur Umwelt“ sollen einerseits über umwelt.info erschließbar sein und andererseits können in diesem Projekt entwickelte Werkzeuge in umwelt.info genutzt werden.

Gemäß Leistungsbeschreibung (Umweltbundesamt, 2020 - a) sind die in den „Daten zur Umwelt“ enthaltenen Daten (vorwiegend Excel-Tabellen) zu berücksichtigen. Geodaten und -dienste sollen über die bestehende Infrastruktur der UBA.gdi genutzt werden.

1.2 Gegenstand

Aus den vorgenannten Zielen sowie der Leistungsbeschreibung (Umweltbundesamt, 2020 - a) ergeben sich die im Projekt zu erfüllenden Aufgaben. Für das vorgesehene intelligente Datenhaltungs- und Datenaufbereitungssystem sollen folgende technische Lösungen konzipiert und umgesetzt werden:

- Der Data-Store ist die zentrale Komponente für die Datenhaltung und muss skalierbar für die zukünftig zu erwartenden Datenmengen ausgerichtet sein. Die Umsetzung geeigneter Datenmodelle ist Bestandteil dieser Komponente.
- Der Data-Input dient für die Dateneingabe bzw. Bereitstellung von Daten durch die Fachseite. Die unterschiedlichen Datenquellen sollen mittels automatisierter Integrationsprozesse in die Datenhaltung überführt und qualitätsgesichert werden.
- Der Data-Explorer soll Funktionalitäten bieten, um Daten zu erkunden und für die Bereitstellung im Internet auszuwählen.
- Unter Data-Output werden flexible Auswertetools, Visualisierungen und Routinen für die Datenbereitstellung gefasst.

Das Vorgehen gliedert sich in die Arbeitspakete Konzeption und Umsetzung.

In der Konzeption werden die Voraussetzungen für die anschließende Umsetzung geschaffen. Die Konzeption gliedert sich in einen fachlichen und einen technischen Teil:

- In der Anforderungsanalyse werden alle fachlichen Bedürfnisse der zukünftigen Anwendenden erhoben (siehe Kapitel 2).
- In der technischen Konzeption wird eine Systemarchitektur entwickelt und es werden die später genutzten Softwarekomponenten festgelegt (siehe Kapitel 3 und 4).

Auf Basis der Konzeption werden die Komponenten des Systems schrittweise umgesetzt und in Betrieb genommen. Da ein agiles Vorgehen geplant ist, ist während der Umsetzung noch eine Feinkonzeption der Datenmodelle, ETL-Prozesse und der Ergebnisdarstellungen enthalten.

2 Anforderungsanalyse

Da die detaillierten Anforderungen an das Data Cube Projekt zum Projektstart noch nicht vollständig beschreibbar waren, sollten zunächst die Anforderungen der Anwendenden erhoben werden. Diese bilden dann im weiteren Verlauf die Grundlage des zu erstellenden Umsetzungskonzepts. Zur Anforderungserhebung wurden Methoden des User-Centered Designs (UCD) in einem agilen Vorgehen eingesetzt. Dieser Ansatz lieferte bereits in der Machbarkeitsstudie des umwelt.info (Umweltbundesamt, 2020 - c) sehr gute Ergebnisse. Im Sinne des UCD erfolgte dieses Vorgehen in enger Zusammenarbeit zwischen dem Fachgebiet (FG) I 1.5 als Auftraggeber und dem Auftragnehmer, sowie im Austausch mit fachlich zuständigen Personen für die 12 Themen der „Daten zur Umwelt“.

Die in den nachfolgenden Kapiteln beschriebenen Anforderungen wurden in mehreren Workshops zu vier Themenkomplexen erarbeitet:

- Redaktioneller Prozess
- Data-Output
- Thematische Gespräche mit den datenhaltenden Stellen
- Technische Anforderungen

Die in den folgenden Kapiteln beschriebenen Anforderungen sind zusätzlich auch als Excel-Tabelle in Anhang A.1 zusammengefasst.

2.1 Redaktioneller Prozess

Unter dem Begriff "Redaktioneller Prozess" werden in diesem Kontext die einzelnen Arbeitsschritte des Fachgebietes I 1.5 zur Veröffentlichung von Daten zur Umwelt beschrieben. Der Prozess beinhaltet den gesamten Vorgang von der Planung einzelner Webseiten-Artikel, über die Kommunikation mit verschiedenen datenhaltenden Stellen und Autoren, dem Zusammenstellen von Daten und Texten, bis hin zur Veröffentlichung auf der UBA-Webseite. Die Betrachtung des Prozesses ist wichtig, um die grundlegenden Arbeitsschritte zur Veröffentlichung von Webseiten-Inhalten zu verstehen und somit Möglichkeiten zur Weiterentwicklung zu konzipieren. Das folgende Kapitel beschreibt die Analyse dieses Prozesses sowie der beteiligten Personengruppen.

2.1.1 Analyse des redaktionellen Prozesses

Zwei interaktive Workshops bildeten den Rahmen für den gewünschten intensiven Austausch und waren zugleich Raum für tiefergehende Gespräche. Die beiden Workshops dienten vor allem dazu, ein gemeinsames Verständnis für den Ist-Zustand des redaktionellen Prozesses sicherzustellen.

Um die beteiligten Akteure des redaktionellen Prozesses zu identifizieren, wurde gemeinsam eine Stakeholder Map erarbeitet. Eine Stakeholder Map ist eine etablierte Methode, um Interessensvertreter und Interessensvertreterinnen (Stakeholder), deren Erwartungen sowie ihre Beziehungen untereinander zu beschreiben. Die Methode wird u.a. von IBM eingesetzt und empfohlen (IBM, o. J.). Die Stakeholder Map soll durch die graphische Darstellung der Beziehungen zwischen Stakeholdern des Fachgebietes I 1.5 das Verständnis der komplexen Beziehungen und Vielschichtigkeit erleichtern. Dabei stellt die Stakeholder Map eine Momentaufnahme dar, die sich ggf. ändern kann und daher im Projektverlauf bei Bedarf nachgepflegt werden muss.

Während des Workshops hatten alle Teilnehmenden die Möglichkeit, ihre Tätigkeiten und Rollen mit Blick auf den Prozess der Veröffentlichung der Daten zur Umwelt zu beschreiben. Die Ergebnisse wurden während der Workshops mit Hilfe des webbasierten Whiteboard-Tools Miro (Miro, o. J.) für alle Teilnehmenden sichtbar festgehalten. Nach dem Workshop wurden die Ergebnisse ausgewertet und die zugehörige visuelle Darstellung optimiert.

Durch die Verwendung einer Stakeholder Map (siehe Tabelle 1) konnten die verschiedenen Projektbeteiligten identifiziert und durch die jeweiligen Rollen und Beziehungen skizziert werden. Der Fokus wurde dabei auf das Fachgebiet I 1.5 gelegt. Grundsätzlich gilt, dass eine Person mehrere Rollen ausführen kann und somit in verschiedenen Stakeholdern enthalten ist. Im Projektverlauf wurden die fett gedruckten Stakeholder als entscheidend für den redaktionellen Prozess identifiziert.

Folgende Stakeholder wurden identifiziert:

Tabelle 1: Auflistung ermittelter Stakeholder mit den jeweiligen Aufgaben

Stakeholder	Aufgaben / Beschreibung
Management	• Change-Management • Projektbegleitung • Management der Prozesse • „Überblick über das Ganze“ • Verzahnung von Weiterentwicklung der Webseite • Leitung Projekt Data Cube
Redaktion	• Verantwortung für einzelne Kapitel
Verwalterischer Überblick und Controlling	• Verwalterischer Überblick • Controlling • Strukturierung der Datenablagen
Qualitätssicherung	• Qualitätssicherung
Visualisierungen	• Bild und Graphikerstellung • Erstellung von interaktiven Diagrammen
Freigabe	• Freigaben vor Veröffentlichung
Datenhaltung und -verarbeitung	• Datenhaltende Fachgebiete (> 50) • Externe Autoren / Autorinnen (BFN, DWD, …) • Externe Datenquellen (Statistisches Bundesamt, …)
Forschung	• Betreuung von Forschungs- und Entwicklungsvorhaben • Bestehende Berichtsprodukte aufeinander zugreifen lassen
Geodaten- und IT-Integration	• Integration von Geodaten in IT-Produkte (fachliche Gestaltung) • Betreuung der IT-Projekte (Weiterentwicklung der Webseite) • Aufbereitung der Geodaten • Schnittstelle FG I 1.7
Spezielle Informationsformate	• Umweltatlas, Stickstoffatlas • Umweltmonitor (statisch und dynamisch) • Numerische Berechnungen
Nutzende der Webseite	• Nutzung der UBA-Webseite

Für die weitere Analyse wurden durch die Methode einer Value Stream Map einzelne Prozessschritte identifiziert und den jeweiligen Stakeholdern zugeordnet.

Eine Value Stream Map (Martin & Osterlin, 2013) ist eine Methode zur Analyse des aktuellen Zustands und zum Entwurf eines zukünftigen Zustands für eine Reihe von Ereignissen, die ein Produkt oder eine Dienstleistung im Rahmen eines bestimmten Prozesses durchläuft. Eine Value Stream Map ist dabei ein visuelles Werkzeug, das alle kritischen Prozess-Schritte sowie Material- und Informationsflüsse darstellt. Während die Stakeholder Map dazu verwendet einzelnen Akteure und deren Beziehungen untereinander zu beschreiben, wird in der Value Stream Map der Fokus auf Prozessschritte der vorher identifizierten Stakeholder gelegt. Die Value Stream Map dient damit dazu, ein gemeinsames Verständnis über Prozesse, wichtige Schnittstellen sowie Rollen und Akteurinnen/Akteure sicherzustellen.

Während des Workshops präsentierten die Mitarbeitenden des UBA die grundlegenden Schritte des redaktionellen Prozesses. Dazu wurden Dokumente wie eine zentrale Excel-Tabelle sowie exemplarische Visualisierungen gezeigt und erklärt. Die Value Stream Map stellt somit einen beispielhaften Prozess zur Bereitstellung der Daten zur Umwelt dar, welcher im Folgenden beschrieben wird.

Der redaktionelle Prozess beginnt durch die Controlling-Stakeholder. Die Organisation aller Artikel (ca. 300) wird mit Hilfe einer Excel Tabelle (Steuerungstabelle) gepflegt. Unter dem Begriff „Artikel“ wird im Folgenden ein Webseiten-Beitrag verstanden, welcher sowohl statische (Texte oder Graphiken) als auch interaktive Elemente (Data-Outputs) beinhalten kann. In der Steuerungstabelle werden alle für die Organisation relevanten Metadaten vorgehalten. Diese sind zum Beispiel:

- Artikel-Name
- Stand im Content-Management-System (CMS)
- Nächste Aktualisierung (zum Beispiel nach Bedarf, bestimmtes Intervall, …)
- Bemerkungen (Austausch mit anderen Autoren, Verfügbarkeit von Daten, Update von anderen Elementen wie Diagrammen bei Aktualisierung)
- Autoren und Autorinnen (intern/extern) / Verantwortlichkeit
- Spiegelreferate (BMU/BMWi)
- Zeitstempel
  - Tag der Anforderung
  - Rücklauf-Termin
  - Eingang
  - BMU/BMWi Abstimmung eingeleitet

Durch eine gute Pflege der Excel Tabelle wird der Überblick über die aktuellen Artikel und die jeweiligen offenen Aufgaben gewährleistet. Zu gegebener Zeit werden neue Daten per E-Mail angefragt und entsprechende Rücklauftermine vereinbart (Stakeholder: Verwalterischer Überblick und Controlling, Datenhaltung und -verarbeitung).

Der nächste Schritt ist die Datenaktualisierung. Neue Daten werden als Excel-Dateien übermittelt. Da Daten bei den Datenlieferanten oft in anderen Systemen vorliegen, müssen diese

zunächst für die entsprechenden Excel-Formatvorlagen konvertiert/angepasst werden. Da es in Excel keinen Änderungsmodus gibt, müssen die Dateien im Anschluss manuell auf Änderungen überprüft werden. Beteiligte Stakeholder in dieser Phase sind: Datenhaltung und -verarbeitung, Verwalterischer Überblick und Controlling, Redaktion.

Anschließend kommt es zur Artikelaktualisierung. Die Artikel werden in Word für das CMS (Drupal) vorgeschrieben, wobei hier der Änderungsmodus verwendet werden kann, um Anpassung nachzuvollziehen. Das Schreiben in Word ist unter anderem notwendig, da nur die Redaktion Zugriff auf das CMS System hat. Die eigentliche Arbeit für Indikatoren wird weiterhin in Excel erledigt. In diesem Schritt werden auch mögliche statische Visualisierungen aktualisiert/erstellt, welche jeweils in verschiedenen Formaten (Excel, PNG, PDF) bereitgestellt werden. Die verschiedenen Formate sind für unterschiedliche Medien (Web-Artikel und Druckversion) benötigt. Die Excel-Tabellen für Indikatoren müssen den UBA-Designvorlagen entsprechen. Abbildungen werden teilweise in mehreren Sprachen entworfen. Nach der Aktualisierung werden die Excel-Dateien wieder an die entsprechenden Quellen zurückgesendet, damit zukünftige Aktualisierung direkt in diese eingearbeitet werden können. Auch die Word-Dateien für Artikel werden via E-Mail ausgetauscht. Auch hier ist eine Fachgebietsübergreifende Kommunikation erforderlich. Bei Bedarf ist an dieser Stelle zusätzlich eine Abstimmung mit dem BMU Spiegelreferat notwendig. Beteiligte Stakeholder sind: Datenhaltung und -verarbeitung, Redaktion, Spiegelreferate / BMU.

Für die Qualitätssicherung werden die Artikel als Word- und Excel-Dateien an die Redaktion übergeben. Zusätzlich zu den Artikel-Inhalten wird das UBA Corporate Design überprüft. Weiterhin wird ermittelt, welche weiteren Elemente (andere Abbildungen, Artikel, …) durch die Aktualisierung angepasst werden müssen. Die Freigabe wird von der Redaktion und bei Bedarf auch von den Spiegelreferaten erteilt. Alle Dateien (Excel, Word, Abbildungen, E-Mail-Kommunikation) werden zentral auf einem Netzwerklaufwerk abgelegt. Alte Versionsstände werden manuell in einen Archiv Ordner kopiert und bei Bedarf auch rückwirkend noch korrigiert. Beteiligte Stakeholder: Qualitätssicherung, Redaktion, Verwalterischer Überblick und Controlling, Visualisierungen, Spiegelreferate / BMU.

Der letzte Schritt ist die Veröffentlichung auf der UBA Webseite. Hierzu werden die Artikel aus den Word-Dateien manuell in die Eingabemaske des CMS kopiert. Interaktive Diagramme für das CMS werden erst in diesem Schritt erzeugt/aktualisiert. Die Daten werden hierzu in eine eigene Eingabemaske der Drupal-Erweiterung Easycharts importiert. Die interaktiven Diagramme können anschließend in den Artikeln verwendet werden. Für die Datensuche der Webseite müssen in der CMS-Eingabemaske noch entsprechende Filter-Zuordnungen eingetragen werden. Die finale Freigabe und Veröffentlichung auf der UBA Webseite erfolgt durch die Chefredaktion. Beteiligte Stakeholder: Verwalterischer Überblick und Controlling, Freigabe, Redaktion, Visualisierungen.

Die verschiedenen Prozessschritte laufen üblicherweise nicht linear ab, sondern beschreiben einen iterativen Prozess, in welchem Daten und Artikel ggf. mehrfach zwischen den Beteiligten ausgetauscht werden. Das Intervall für eine Artikelaktualisierung variiert zwischen wenigen Tagen und ein bis zwei Jahren.

2.1.2 Fazit

Die folgenden Punkte beschreiben Elemente des redaktionellen Prozesses die aus Sicht des Auftragnehmers besonders wichtig erscheinen und/oder Möglichkeiten zur Verbesserung bieten.

Die Analyse des redaktionellen Prozesses zeigt, dass es einige klar definierte Prozessschritte gibt, für welche häufig eine intensive Kommunikation mit verschiedenen Stakeholdern notwendig ist. Die Kommunikation ist hierbei fachgebietsübergreifend und beinhaltet verschiedene Rollen. Da diese Abstimmungen durch die verschiedenen Aktualisierungszyklen der Berichte sehr unterschiedlich ausfallen, ist eine klare Dokumentation des aktuellen Standes sowie der zukünftigen Aufgaben sehr wichtig. Als zentrales Element dient hier aktuell eine Excel-Tabelle mit Übersicht über alle Artikel und entsprechenden Metainformationen.

Viele Schritte sind durch manuelle Tätigkeiten geprägt. Als Beispiele sind hier der Austausch von Dokumenten via E-Mail, die Dateiablage, Bereitstellungen von Exporten in verschiedenen Formaten und das Kopieren der Berichte/Daten in die CMS Eingabemaske zu nennen.

Die Themen Qualitätssicherung und Freigabe sind von hoher Bedeutung. Alle Informationen (Artikel, Abbildungen, Daten etc.) die veröffentlich werden sollen, müssen von den jeweiligen beteiligten (Redaktion, Spiegelreferate) sowie der Chefredaktion freigegeben werden.

Der Austausch von Daten (Kennzahlen, sowie Berichte) erfolgt überwiegend durch den Versand von Dateien via E-Mail. Für den Austausch werden Word- und Excel-Dateien verwendet, da diese eine einfache Bearbeitung durch alle Beteiligten ermöglichen. Vor allem für Word-Dateien ist der Änderungsmodus besonders wichtig. Durch die lokale Bearbeitung von Dateien durch die einzelnen Mitarbeitenden erzeugt das Verfahren jedoch gleichzeitig eine hohe Redundanz. In der Regel sind solche Verfahren potenziell Fehleranfällig, da zeitgleich Dateien an verschiedenen Orten bearbeitet werden können und ein Zusammenführen stets durch manuelle Arbeit erfolgt. Zusätzlich lösen Änderungen an den Daten einige manuelle (zeitintensive) Schritte aus, bis eine erneute Veröffentlichung stattfinden kann. Ändern sich Kennzahlen in der Datenquelle müssen zum Beispiel die folgenden Schritte durchgeführt werden: Export der Kennzahlen in das Excel Format, Übermittlung via E-Mail an die Redaktion, Anpassung der Artikel im CMS, Anpassung der statischen Abbildungen, Export der Abbildungen in verschiedene Formate, Kopieren des Artikels in das CMS, Importieren der Daten für interaktive Diagramme, Veröffentlichung, Archivierung aller relevanten Dateien.

2.2 Data-Output und Data-Explorer

Ein zentraler Aspekt des Data Cube Projektes ist die Darstellung der Daten auf der Webseite des UBA. Im Gespräch zum Themenkomplex Data-Output wurden daher verschiedene Beispiele existierender Datenportale besprochen und allgemeine Anforderungen diskutiert.

2.2.1 Diskussion von Beispielen

In diesem Kapitel werden die vorgestellten Beispiele erläutert und mit Bezug auf den Data Cube in Relation gesetzt. Bei den vorgestellten Beispielen handelt es sich um konkrete Umsetzungen von Datenportalen verschiedener Organisationen bzw. Firmen, welche als Diskussionsgrundlage für die Anforderungsanalyse dienten.

Die folgenden Beispiele wurden während des Workshops betrachtet:

- ROSYS (Deutsche Rohstoffagentur, Bundesanstalt für Geowissenschaften und Rohstoffe, o. J.) – ist ein Daten Portal der Deutschen Rohstoffagentur (DERA) und der Bundesanstalt für Geowissenschaften und Rohstoffe (BGR) und soll Unternehmen sowie der Öffentlichkeit helfen, die globale Rohstoffsituation besser zu verstehen. Anhand des ROSYS können Filtermöglichkeiten, Interaktive Graphiken, Download Optionen sowie die Kombination von Information und Karte demonstriert werden. Das Portal ist zudem gemeinsam mit con terra entwickelt worden.

- Carbon Disclosure Project (CDP) (Carbon Disclosure Project - CDP, o. J.) – ist eine gemeinnützige Organisation, die Unternehmen und Behörden in der Minderung ihrer Auswirkungen auf die Umwelt unterstützt. Das Ziel ist, den Klimawandel bestmöglich zu bremsen. Das Datenportal demonstriert eine interessante Einstiegseite sowie die Komplexität einer großen, heterogenen Datenstruktur.
- Australian Early Development Census (AEDC) (Australian Early Development Census - AEDC, o. J.) – ist die Datensammlung einer australischen Regierungsinitiative, die Kommunen dabei hilft, die Situation von Kleinkindern und Familien nachhaltig zu unterstützen. Dazu werden alle drei Jahre Daten auf nationaler Ebene erhoben. Das Portal von AEDC zeigt die Möglichkeiten von Filtern und Disaggregationen, einfache graphische Umsetzungen sowie Downloadmöglichkeiten.
- United Nations Data Portal Population Division (United Nations - UN, o. J.) – die Vereinten Nationen stellen über das Portal Daten zu weltweiten demographischen Indikatoren bereit. Nutzende können anhand von Ländern und diversen anderen Kriterien die Daten durchsuchen. Zusätzlich zum Web-Portal sind die Daten zudem noch über eine API abrufbar. Das Data Portal stellt tabellarische Daten sowie die dazugehörige, interessante Visualisierungsoptionen bereit und bietet Filter und Download Funktionen und eine zeitliche Aufarbeitung der Daten.
- Emissions Database for Global Atmospheric Research (EDGAR) (Europäische Kommission, o. J. - a) – ist eine unabhängige, globale Datenbank zu anthropogenen Emissionen von Treibhausgasen und Luftverschmutzung auf der Erde. Über die Webseite können Nutzende verschiedene Dashboards zu den Themen: „Country Fact Sheet“, „Greenhouse Gases Emissions and Climate“ und „Air and Toxic pollutants“ interaktiv betrachten. In den Dashboards werden Daten als interaktive Diagramme dargestellt, welche durch verschiedene Dropdown-Menüs gefiltert, aggregiert und disaggregiert werden können.
- Organisation for Economic Co-operation and Development (OECD) (Organisation for Economic Co-operation and Development - OECD, o. J.) – ist eine internationale Organisation, die gemeinsam mit Bürgern, Entscheidungsträgern und Regierungen versucht, Lösungen für soziale, wirtschaftliche und ökologische Herausforderungen auf datenbasierten Erkenntnissen zu finden. In OECD Data werden Dashboards und Visualisierungen zu den verschiedenen Datenthemen der Organisation angeboten.
- Food and Agriculture Organization (FAO) (Food and Agriculture Organization of the United Nations - FAO, o. J.) – ist eine Organisation der Vereinten Nationen zur Bekämpfung von Hunger. In FAOSTAT können verschiedene Indikatoren dargestellt werden. Dazu werden interaktive Dashboards aber auch Möglichkeiten zum Vergleichen von Datensätzen bereitgestellt.

Bei allen ausgewählten Beispielen können Datensätze als interaktive Diagramme und als Tabelle dargestellt werden. Je nach Datensatz ist es teilweise auch möglich, eine kartenbasierte Darstellung zu wählen. Die reine Darstellung als interaktive Diagramme wird als Standardfunktionalität angesehen und muss im Data Cube Projekt verfügbar sein.

Vor allem die Konfigurierbarkeit von Darstellungen zur Exploration der Daten durch den Nutzenden unterscheidet sich stark zwischen den verschiedenen Beispielportalen. Die größte Vielfalt an Funktionalitäten bietet das CDP Portal. Nutzende können Daten sortieren, aber auch Roll-Ups konfigurieren, um Datensätze anhand verschiedener Tabellenspalten zu aggregieren. Dies bedeutet, dass man als Nutzender eigenständig einfache Berechnungen auf Basis der Daten

durchführen kann. Je nach Datentyp kann man zum Beispiel einzelne Tabellenspalten berechnen (Mittelwert, Summe, Maximum, Minimum) oder einzelne Werte zählen und diese nach einer weiteren Spalte gruppieren. Diese Funktionalitäten sind für den Data Cube gewünscht, allerdings ist die Darstellung im CDP Portal schwierig zu verstehen und nur für erfahrene Nutzende sinnvoll. Die Darstellung von interaktiven Diagrammen ist in diesem Portal ebenfalls sehr komplex und erlaubt dem Nutzenden viele verschiedene Möglichkeiten zur Visualisierung der Daten. Graph-Typen (Linien-, Balken-, Kreisdiagramm, etc.) müssen/können ebenso wie die einzelnen Dimensionen und Messwerte (siehe Kapitel 3.1.1.2) für jeden Datensatz individuell gewählt werden. Vordefinierte Diagramme sind hingegen nicht vorhanden.

Die Portale von EDGAR, AECD, FAOSTAT und ROSYS sind hier wesentlich simpler gehalten. Die Darstellungsarten sind vorgegeben und lediglich die Inhalte können durch vorkonfigurierte Selektionen (zum Beispiel durch Dropdown-Listen) angepasst werden. In den Portalen kann jeweils nach Regionen gefiltert werden. Mit Ausnahme von EDGAR kann auch eine zeitliche Selektion durchgeführt werden. Diese Darstellungsarten wurden von den Teilnehmenden im Workshop wesentlich intuitiver aufgefasst. Die beste Darstellung aus Sicht des Auftraggebers hat das OECD Portal. Einzelne Datensätze werden bereits vorkonfiguriert dargestellt, wobei die einzelnen sogenannte Perspektiven der Daten aktiviert/deaktiviert werden können. Unter Perspektiven sind im OECD Portal verschiedene Sichten auf die Datensätze beschrieben. In vielen Fällen gibt es eine Perspektive zur Beschreibung von aggregierten Werten (zum Beispiel Summe aller möglichen Kategorien) sowie einzelne Perspektiven auf die jeweiligen Kategorien der Datensätze. Dadurch können Nutzende durch Umschalten der Perspektiven Disaggregationen durchführen, um zu verstehen, aus welchen Einzelelementen die jeweilige Aggregation besteht. Zusätzlich ist eine zeitliche Filterung möglich und einzelne Datensätze können farblich hervorgehoben werden. Sehr positiv bewertet wurde auch die Möglichkeit, die Einheit der jeweiligen Darstellung auszuwählen. Je nach Datensatz sind einige Konfigurationsmöglichkeiten jedoch ausgegraut und können nicht verwendet werden. Eine Anpassung der Visualisierungsart (zum Beispiel Graph-Typen) ist nicht möglich.

Die Portale von FAOSTAT und EDGAR Portal bieten Beispiele für Dashboard. Bei beiden wurde die Übersichtlichkeit und die einfach zu bedienenden Filtermöglichkeiten positiv bewertet. Allerdings fehlen bei EDGAR Erklärungen (zum Beispiel Querverweise) über die Daten. Die Art der Visualisierungen kann in beiden Dashboards nicht angepasst werden.

Im ROSYS Portal wurden die „Touren“ positiv bewertet. Diese bieten thematische Einstiegspunkte in die Anwendung, wobei der Nutzende nach dem Starten der Tour eine interaktive Erklärung der Webseite zur Erläuterung der verschiedenen Funktionalitäten erhält. Vor allem für neue Nutzende bietet diese Funktion einen guten Einstieg in die Anwendung. Negativ bewertet wurde, dass die Touren nicht logisch gruppiert sind.

In FAOSTAT ist weiterhin eine Funktionalität zum Vergleichen von Datensätzen vorhanden. Nutzende können für eine wählbare Zeitspanne verschiedene Datensätze durch Filter auswählen, die dann in einem interaktiven Linien-Diagramm zeitgleich dargestellt werden. Dadurch können die zeitlichen Verläufe verschiedenster Indikatoren miteinander vergleichen werden.

Exportfunktionalitäten sind ebenfalls in fast allen Anwendungen möglich, wobei sich die Wahl der Formate jedoch stark unterscheidet. Während das OECD Portal nur CSV als Format anbietet, können im CDP Portal eine Vielzahl von verschiedenen Formaten (CSV, JSON, RDF, RSS, TSV, XML) verwendet werden. Die konkreten Diagramme werden nur teilweise als Export angeboten, wobei hier jeweils statische Abbildungen (PDF, PNG, JPG) verwendet werden.

Die Portale des OECD, CPD, FAOSTAT und der United Nations besitzen zusätzlich zur reinen Darstellung der Daten auch eine Katalog-Funktion. Nutzende können entweder über die Suchfunktion oder durch Auswahl verschiedener Rubriken verschiedene Datensätze auffinden. Für die einzelnen Einträge werden dabei auch entsprechende Metadaten aufgelistet. Diese sind unter anderem Titel, Beschreibung, Quelle und Datum. Während das CDP direkt auch technische Metadaten (Beschreibung der Attribute mit Datentypen) auflistet, verweist das OECD Portal zusätzlich auf verwandte Publikationen. In FAOSTAT werden die Metadaten für einzelne Indikatoren sehr ausführlich beschrieben. Die Metadaten beinhalten Informationen über Ansprechpartner, aber auch Einheiten, wesentliche Charakteristika, mögliche Fehlerquellen und einige weitere Kategorien.

2.2.2 Anforderungen an Data-Explorer und Data-Output

Auf Basis der vorgestellten Beispiele und den diskutierten Fragen ergeben sich die folgenden Anforderungen an die Komponenten des Data-Output und Data-Explorer.

2.2.2.1 Data-Explorer

Unter dem Begriff Data-Explorer wird eine interne Komponente für die folgenden drei Aufgabengebiete zusammengefasst:

- Auflistung aller Inhalte des Data-Stores (s. Kapitel 2.4) und Zusammenstellen verschiedener Datensätze zu thematischen Cubes
- Interaktive Darstellung der Daten (Exploration)
- Konfiguration verschiedener Data-Outputs (s. Kapitel 2.2.2.2).

Die detaillierteren Anforderungen werden im Folgenden beschrieben.

Innerhalb des Data-Explorers soll es möglich sein, die gesamte Datenhaltung zu betrachten. Dies bedeutet, dass eine Auflistung aller Datensätze inklusive der entsprechenden Metadaten sowie Anzeige der konkreten Inhalte möglich sein muss. Hierfür soll es sowohl tabellarische als auch graphische (interaktive Diagramme) Möglichkeiten geben. Die Darstellung von Diagrammen soll frei konfigurierbar sein, wodurch verschiedene Darstellungsmethoden (zum Beispiel Diagramm-Typen) getestet werden können. Dieser Überblick wird unter anderem benötigt, um anschließend Datensätze zu thematisch sinnvollen Cubes zu kombinieren. Unter einen Cube wird in diesem Kontext eine Ansammlung von Datensätzen bezeichnet, wobei sich aus jedem Datensatz Dimensionen für den Cube ergeben. Eine detailliertere Erklärung zu Dimensionen und Cubes findet sich in Kapitel 3.1.1.2. Datensätze, die für einen Cube ausgewählt werden, sollen später im Data-Output zur Exploration bereitgestellt werden. Über den Data-Explorer soll es möglich sein, alle Cubes aufzulisten und die entsprechend verlinkten Datensätze anzuzeigen. Zusätzlich soll es für jeden Datensatz möglich sein, alle Cubes aufzulisten, in denen der entsprechende Datensatz enthalten ist.

Zusätzlich soll der Data-Explorer verwendet werden, um verschiedene Data-Outputs zu definieren. Eine detaillierte Beschreibung der Data-Outputs erfolgt in Kapitel 2.2.2.2. Inhaltlich gewünschte Konfigurations-Optionen für Data-Outputs sind: Anpassung der auswählbaren Dimensionen, verfügbare Einheiten, mögliche Filterungen auf den Daten, Download-Optionen, Verknüpfungen zu anderen Datensätzen, sowie Metainformationen. Je nach Data-Output soll zusätzlich ausgewählt werden können, welche Werkzeuge dem Nutzenden zur Verfügung gestellt werden. Die Möglichkeiten sollen von der vollen Funktionalität (inkl. Filterungen, Aggregationen, etc.) bis hin zu einfachen, fest definierten Diagrammen konfigurierbar sein. Die

Konfiguration soll in einer einfachen Web-Maske möglich sein. Denkbar ist zunächst eine volle Darstellung aller Möglichkeiten, welche durch die Redaktion weiter eingegrenzt werden könnte (zum Beispiel durch Checkboxen an den einzelnen Werkzeugen). Eine andere Option wäre die Verwendung von vorgefertigten Profilen zum Aktivieren bzw. Deaktivieren verschiedener Data-Output Möglichkeiten. Auch eine Kombination aus beiden Möglichkeiten wäre denkbar.

Zusätzlich zu den einzelnen Data-Outputs soll es möglich sein, mehrere Elemente zu einem Dashboards zusammenzufassen. Dafür sollen mehrere Outputs konfiguriert und entsprechend räumlich angeordnet werden können.

Komplexe statistische Analysen werden in den meisten Fällen in externen Expertenprogrammen durchgeführt. Hierzu müssen die Daten aus dem Data-Store verwendet werden können. Dazu soll es möglich sein, Daten über eine API bereitzustellen, um diese programmatisch auch in Skripten verwenden zu können. Zusätzlich soll ein dateibasierter Export der Daten in einem maschinenlesbaren Format (zum Beispiel JSON, CSV, NetCDF) möglich sein.

Der Zugriff auf den Data-Explorer muss für verschiedene Nutzergruppen möglich sein. Eine genaue Definition der Zugriffsrechte steht derzeit noch aus. Grundsätzlich sollen Anpassungen jedoch nur von der Redaktion vorgenommen werden können, während andere Nutzende den Data-Explorer primär zur Exploration der Daten sowie durch Zugriff über die API benutzen sollen.

Der Data-Explorer kann außerhalb des UBA CMS implementiert werden und hat kein spezifisches CMS zur Voraussetzung.

2.2.2.2 Data-Output

Unter dem Begriff Data-Output wird die Darstellung der verschiedenen Datensätze/ Cubes auf der Webseite des UBA für Endanwender beschrieben. Data-Outputs sind ein Oberbegriff für:

- Elemente, welche in bestehende Webseiten integriert werden können, um Daten graphisch anzuzeigen
- einfache interaktive Diagramme oder Tabellen
- komplexere interaktive Diagramme zur explorativen Betrachtung verschiedener Datensätze
- oder kombinierte Dashboards, welche aus mehreren einzelnen Data-Outputs bestehen können. Um verschiedene Anwendungsfälle abbilden zu können, sollen die Data-Outputs vollständig konfigurierbar sein.

Data-Outputs (Diagramme, Tabellen und Dashboards) müssen direkt in Artikel, oder als eigenständige Seiten auf der UBA-Webseite (Drupal CMS) eingebunden werden können. Die Implementierung soll dabei jedoch möglichst CMS-offen (unabhängig von dem bestehenden Drupal System) erfolgen. Es soll geprüft werden, ob Inhalte zum Beispiel als responsive iFrames oder vergleichbares eingebettet werden können. Auch eine Verlinkung zu externen Seiten muss evaluiert werden.

Funktionalitäten, die nicht CMS-offen implementiert werden, sind als Drupal-Modul (unter Beachtung der Drupal Code Konventionen) zu entwickeln und vollständig lauffähig in das CMS der UBA Webseite zu integrieren.

Alle Data-Outputs sollen auf verschiedenen Display Größen (Mobilgeräte und größere Bildschirme) gut angezeigt werden (Responsive-Design).

Im Vergleich zum Data-Explorer (s. Kapitel 2.2.2.1) sollen im Data-Output nur die zuvor konfigurierten Möglichkeiten zur Verfügung gestellt werden. Durch die Redaktion soll vorgegeben werden, welcher Output welche Funktionalitäten benötigt, um je nach Zielgruppe eine gute Nutzererfahrung zu gewährleisten.

Data-Outputs sollen entweder vorkonfiguriert nutzbar, oder durch den Anwendenden angepasst werden können. Dafür muss eine Visualisierung von Daten möglich sein, bei denen eine feste Ansicht auf die Daten und eine feste Darstellungsform durch die Redaktion vorgegeben wird. Zusätzlich soll die Darstellungsform der Abbildungen durch den Nutzenden frei gewählt werden können. Gewünschte Darstellungsformen sind:

- Liniendiagramme
- Balkendiagramme
- Tortendiagramme
- Baumdiagramme
- Abweichungen
- Korrelationen und Streudiagramme
- Häufigkeitsverteilungen
- Nominale Vergleiche (zum Beispiel Blasendiagramme oder Heatmaps)
- Fluss und Sankey-Diagramme
- Netzwerkdiagramme

Weiterhin müssen Daten als Tabelle dargestellt werden können, wobei es innerhalb der Tabelle möglich sein soll, Datensätze zu filtern und zu sortieren.

Zur weiteren Exploration der Daten sollen verschiedene interaktive Funktionen bereitgestellt werden. Nutzende sollen die Möglichkeit haben, Datensätzen zu aggregieren/disaggregieren, oder vorberechnete und entsprechend verlinkte Aggregationen/Disaggregationen anzuzeigen. Aggregationen beschreiben Zusammenfassungen von Daten. Für Zusammenfassungen werden üblicherweise Gruppierungen über bestimmte Dimensionen durchgeführt und die entsprechenden Werte anhand einer Funktion errechnet. Einfache Aggregations-Funktionen sind zum Beispiel Summen, Mittelwerte oder das Zählen von Werten. Wie in Kapitel 2.3 beschrieben, sind einige Aggregationen jedoch komplexer, wodurch diese nicht über die Webseite berechnet werden dürfen, sondern im Vorraus durch die entsprechende datenhaltende Stelle bereitgestellt werden müssen.

Zeitreihen sollen weiterhin zeitlich eingeschränkt werden können, wobei hier eine Filterung durch ein minimales und maximales Datum möglich sein soll.

Einheiten sollen dynamisch umgerechnet werden können. Im Data-Output könnten hierzu Daten direkt im Browser umgerechnet werden (zum Beispiel m auf Km). Hierzu müssten Umrechnungsformeln entsprechend definiert sein. Falls keine einfache Umrechnung möglich ist, könnte ein Referenzdatensatz verlinkt werden, in dem die entsprechende Berechnung schon beim Einspielen der Daten in den Data-Store vorgenommen wurde.

Nutzenden soll es zusätzlich möglich sein, Datensätze, die zu einem Thema gehören, zu erkunden. Das heißt, dass zusammengehörige Daten über die Webseite auffindbar gemacht

werden und diese miteinander kombiniert/verglichen werden können. Dazu sollen Daten, falls möglich, in einem gemeinsamen Diagramm überlagert werden. Alternativ dazu, könnte zwischen verschiedenen Data-Outputs gewechselt werden. Ein Beispiel für solche Verknüpfungen sind Datensätze des gleichen Themas mit verschiedenen Aggregationsebenen (zum Beispiel Aufsummierte Werte nach Tagen oder Jahren).

Weiterhin sollen Daten und Abbildungen in unterschiedlichen Formaten zum Download angeboten werden. Für Daten sollen maschinenlesbare Formate wie JSON, CSV, NetCDF verfügbar sein. Abbildungen sollen als statische Bilder (PNG, JPG) exportierbar sein.

Bei der Gestaltung von Nutzeroberflächen muss das Corporate Design des UBA berücksichtigt werden.

Data-Outputs sollen zusätzlich auch in die Datensuche des UBA eingebunden werden können.

2.3 Thematische Gespräche mit den datenhaltenden Stellen

Wie in den vorherigen Kapiteln beschrieben, soll der Data Cube Daten verschiedenster Themen zur Exploration durch die Nutzenden bereitstellen können. Da diese Daten durch verschiedene Stellen vorgehalten werden, existieren sehr heterogene technische und organisatorische Strukturen, welche für den Datenimport in den Data Cube betrachtet werden müssen. Hierzu wurden mehrere Gespräche mit den einzelnen datenhaltenden Stellen geführt, um sowohl Datenquellen als auch Austauschmöglichkeiten genauer zu diskutieren. Der Ablauf der Gespräche wird im Folgenden genauer erläutert.

Im Projektverlauf wurde entschieden, dass zunächst keine weiteren Gespräche mit datenhaltenden Stellen für die Konzeption notwendig sind, auch wenn nicht mit allen verschiedenen Stellen gesprochen wurde. Das entsprechende Projektbudget soll stattdessen während der Implementierungsphase für Detailgespräche verwendet werden.

2.3.1 Methodik

Um die Gespräche mit den verschiedenen datenhaltenden Stellen vergleichbar zu gestalten, wurde ein einheitlicher Gesprächsleitfaden entwickelt. In den Gesprächen stellte das UBA FG I 1.5 zunächst den Teilnehmenden das Projekt vor, um ein gemeinsames Zielverständnis zu entwickeln. Hierzu wurde die Motivation für den Data Cube sowie der Ist-Zustand des aktuellen redaktionellen Prozesses erläutert. Zusätzlich wurden die einzelnen Komponenten des Data Cubes kurz beschrieben.

Anschließend ordnete der Auftragnehmer die Gespräche für die Teilnehmenden in den Gesamtkontext des Projektes und den groben Zeitplan ein. Es folgte eine kurze Erklärung zum Datenimport mittels ETL Software am Beispiel FME Desktop.

Die Teilnehmenden der datenhaltenden Stellen hatten anschließend jeweils Zeit, ihre Daten selbst zu erläutern. Es folgte die Vorstellung der Gesprächs-Leitfragen mit einer offenen Diskussionsrunde.

Die folgenden Leitfragen wurden verwendet:

- Welche Datenquelle kann für den Import verwendet werden?
- Wie wird diese Datenquelle erzeugt?
  - Prozess (Manuell vs. Automatisch)
  - Was ist die eigentliche Quelle/Datenherkunft?

- Wie werden die Daten technisch vorgehalten?
  - Struktur, Format, Dimensionen, Einheiten, Hierarchien
- Gibt es Sektor-Abgrenzungen?
- Welche Aktualisierungszyklen/-abläufe gibt es?

- Wie wird die aktuelle Austausch-Exceltabelle befüllt?
- Gibt es Verknüpfungen mit anderen Datensätzen?
- Einheiten
  - Welche Einheiten liegen vor?
  - Dürfen Umrechnungen im Data-Output durchgeführt werden?
- Aggregationen
  - Gibt es Aggregationen innerhalb der Datensätze?
  - Wie werden diese Aggregationen berechnet?
  - Können Aggregationen in den Data-Outputs berechnet werden oder liegen komplexere Algorithmen vor?
- Gibt es grundlegende Unterschiede in der Datenbereitstellung zwischen den verschiedenen Datensätzen des FGs?

Zusätzlich zu den Leitfragen wurde vor dem ersten Gespräch durch den Auftragnehmer eine Excel-Tabelle zur Beschreibung der zuvor übermittelten Datensätze vorbereitet. Diese wurde im Voraus als Vorbereitung auf den Termin durch den Auftragnehmer ausgefüllt und während der Gespräche weiter vervollständigt. Die Excel-Tabellen sind im Anhang A.2 zusammengefasst.

Die Gespräche hatten jeweils eine Dauer von zwei Stunden.

2.3.2 Ergebnisse

2.3.2.1 Fachthemen: Private Haushalte und Konsum sowie Umwelt und Wirtschaft

In dem Gesprächstermin (30.06.2021) zum Thema Private Haushalte und Konsum sowie Umwelt und Wirtschaft waren Teilnehmende aus dem Fachgebiet I 1.4 (Umwelt und Wirtschaft) sowie dem Fachgebiet I 1.6 (Klimaanpassung) anwesend. Im Folgenden werden die wesentlichen Punkte zusammengefasst.

Die Daten der verschiedenen Fachgebiete liegen überwiegend als Excel-Dateien vor, welche zu einem großen Teil vom Statistisches Bundesamt (DESTATIS) bezogen werden. Die Bearbeitung, insbesondere die derzeitige Aufbereitung für die Übermittlung zu DzU, wird dabei manuell durchgeführt. In einzelnen Ausnahmen werden die bestehenden Excel-Templates auch extern (zum Beispiel durch den Deutschen Wetterdienst (DWD)) befüllt. Automatische Prozesse zur Datenlieferung an das FG I 1.5 existieren derzeit nicht. Daher wäre eine Anpassung der Datenübermittlung (zum Beispiel durch ein anderes Excel-Template) denkbar, wenn dadurch die maschinelle Verarbeitung vereinfacht werden könnte. Es wurde jedoch auch diskutiert, dass nicht alle Rohdaten ohne entsprechende Aufbereitung veröffentlich werden können. Gründe hierfür sind sowohl eine notwendige Qualitätssicherung als auch sensible Daten, welche erst bereinigt werden müssten.

Insgesamt gibt es wesentlich mehr Daten, als derzeit über die Webseite bereitgestellt werden. Forschungsnehmer könnten grundsätzlich wesentlich mehr Daten für den Data Cube liefern, der initiale Aufwand ist jedoch sehr groß, da jeweils ein Artikel zu den Daten verfasst werden soll.

In vielen Datensätzen existieren Klassifizierungen in verschiedenen Kategorien (zum Beispiel Wirtschaftskategorien). Als potenzielle Risiken des Projekts wurden Verknüpfungen von Dimensionen identifiziert, die aus fachlicher Sicht nicht miteinander verglichen werden können. Verknüpfungen von Datensätzen sollten daher nur durch die Redaktion konfiguriert werden können und sind nicht durch Endanwendenden auf der Webseite frei wählbar. Zusätzlich müssten Kategorien innerhalb der Datensätze klar definiert sein, um eine Vergleichbarkeit zu gewährleisten. Kategorien werden derzeit teilweise aus zentralen Registern (zum Beispiel DESTATIS, Klassifikation der Wirtschaftszweige, Ausgabe 2008) abgeleitet und für konkrete Zusammenstellungen umbenannt, weshalb die Kategorien innerhalb der Daten nicht ausreichend sind. Entsprechende Quellen für die Kategorien und Zusammenhänge müssten in den Metadaten erfasst werden. In einigen Fällen werden Klassifikationen zur Veröffentlichung angepasst (zum Beispiel Zusammenlegung von Gütergruppen zu Bedarfsfeldern beim Konsum privater Haushalte zur Darstellung der direkten und indirekten CO2-Emissionen und des Energieverbrauchs). In diesen Fällen soll die Zuordnung der Klassifikationen ebenfalls über die Metadaten ersichtlich sein.

Innerhalb der Datensätze liegen die Daten oft in verschiedenen Aggregationen vor. Dabei kommt es teilweise auch zu mehrstufigen Berechnungen (zum Beispiel zuerst Summenbildung über eine Kategorie, dann Berechnung von Prozentwerten). Ob Daten auch dynamisch im Data Cube berechnet werden könnten, kann nicht global entschieden werden. Die Entscheidung sollte im Einzelfall pro Datensatz abgestimmt werden. Es wäre jedoch wünschenswert, einzelne Berechnungsschritte im Data Cube zu hinterlegen und dynamisch zu berechnen, oder die bereits berechneten Werte entsprechend zu verlinken.

Die Fragestellung, wie mit Daten aus externen Quellen im Data Cube umgegangen werden soll, ist derzeit noch offen und konnte noch nicht geklärt werden. Ein Beispiel hierzu sind die Genesis-Datenbank von DESTATIS, aber auch vom Statistischen Bundesamt veröffentlichte Excel-Tabellen.

Insgesamt wurde das Projektvorhaben des Data Cube von den Teilnehmenden sehr positiv bewertet. Das Vorhaben kann nach Einschätzung der Teilnehmenden sowohl intern Prozesse vereinfachen, um in Zukunft mehr Daten bereitstellen zu können, als auch die Außendarstellung auf der Webseite verbessern.

2.3.2.2 Fachthema: Boden und Wasser

In dem Gesprächstermin (20.07.2021) zum Thema Boden und Wasser waren Teilnehmende aus dem Fachgebiete II 2.1 (Wasser und Boden), dem Fachgebiet II 2.7 (Bodenzustand, Bodenmonitoring), sowie externe Auftragnehmer der Firma ENDA GmbH & Co. KG anwesend. Im Folgenden werden die wesentlichen Punkte zusammengefasst.

Während des Termins wurde das F&E Projekt Fachinformationssystem (FIS) Wasser und Boden durch die Teilnehmenden von FG II 2.1 und FG II 2.7 vorgestellt. In dem Projekt soll ein einheitliches Informationssystem für Wasser- und Bodendaten im UBA entwickelt werden. Hierzu wird das Datenmanagement fachlich und technisch komplett neu aufgestellt. Derzeit liegen die Daten in fünf unterschiedlichen Datenbanken, intern wie extern, vor. Durch das Projekt sollen Prozesssteuerung von Datenbereitstellungen, Qualitätssicherung, Abfrage sowie Aggregationen verbessert werden. Das Teilvorhaben Data Cube ist bereits explizit als externer Anknüpfungspunkt genannt. Innerhalb des FIS Wasser und Boden soll die Datenbereitstellung

über ein Data-Warehouse System realisiert werden. Für den Data Cube sollen nur qualitätsgesicherte und bereits aggregierte Daten inkl. Metadaten bereitgestellt werden. Als Schnittstelle zur Datenübernahme in den Data Cube wäre eine REST Schnittstelle denkbar. Da das Data-Warehouse jedoch noch nicht final konzipiert ist, muss eine weitere Abstimmung zu den technischen Begebenheiten im Jahr 2022 stattfinden. Das geplante Projektende für das FIS Projekt ist Ende 2023.

Durch die Teilnehmenden wurde berichtet, dass alle Daten in den entsprechenden Datenbanken vorliegen. Diese werden vorher aus Excel/CSV-Dateien aufbereitet, die durch die Bundesländer in der Regel jährlich aktualisiert/bereitgestellt werden (Bodendaten nur alle 3-4 Jahre). Die Zustandsdaten zu Gewässern werden getrennt zu den Themen Grundwasser, stehende Gewässer, fließende Gewässer und Meere übergeben. Die gelieferten Datenstrukturen sind dabei sehr heterogen und können sich je nach Lieferung in Struktur und Umfang unterscheiden. Im Gespräch wurde das am Beispiel von Zustandsdaten zu stehenden Gewässern (Seen) demonstriert.

Es wurde ebenfalls diskutiert, dass mehr Daten geliefert werden könnten, wenn der initiale Aufwand (zum Beispiel Formatierung der Excel-Templates und Verfassen von Artikeln) geringer wäre. Grundsätzlich soll jedoch auch beachtet werden, inwieweit eine doppelte Bereitstellung von Daten vermieden werden kann, da es bereits Datenlieferungen für Wasser-DE (Bundesanstalt für Gewässerkunde, o. J.) gibt.

Die Implementierung des FIS Wasser und Boden erfolgt über einen externen Dienstleister.

Insgesamt wurde die grundlegende Idee des Data Cube Projekts von allen Teilnehmenden als sehr positiv bewertet, und die Vorteile des Data Cube wurden sehr früh im Gespräch erkannt.

2.3.2.3 Fachthema: Emissionen

In dem Gesprächstermin (21.07.2021) zum Thema Emissionen waren Teilnehmende aus dem Fachgebiet V 1.6 (Emissionssituation) anwesend. Im Folgenden werden die wesentlichen Punkte zusammengefasst.

Innerhalb des FG V 1.6 wird die Emissionsdatenbank mit der Software Mesap der Firma Seven2One eingesetzt, um Daten aus verschiedensten Quellen (Excel, CSV, WebServices) zu speichern und zu verarbeiten. Es liegen Zeitreihen ab 1990 vor, welche in beliebigen Aggregationen/Disaggregationen ausgegeben werden können. Als Ausgabe wird in der Regel Excel oder XML verwendet. Innerhalb von Mesap können generische Datenräume mit Dimensionen als Bäume definiert werden. Durch die Bäume werden die Datenstrukturen und die darin enthaltenen Aggregationslevel der einzelnen Datensätze beschrieben. Dadurch können Ausgaben relativ frei für den Data Cube oder andere Ziele definiert werden. Innerhalb der Bäume werden sowohl Zeitreihen als auch einzelne Daten verschlagwortet. Zusätzlich können entlang der Baumstrukturen Berechnungen über alle Aggregationsebenen hinweg durchgeführt werden. Wichtig ist hierbei, dass Kategorien und Ebenen von ihrem jeweiligen Kontext abhängig und nicht immer direkt vergleichbar sind. Für den Data Cube ist es wichtig, diese Kategorien zu verstehen. Für einen Vergleich mit anderen Daten muss jeweils ein kleinster gemeinsamer Nenner in den Aggregationsebenen gefunden werden. Zusätzlich ist es wichtig, dass die Metadaten über die Klassifizierungen gut gepflegt sein müssen.

Welche Daten in Zukunft über DzU bereitgestellt werden sollen, muss noch entschieden werden. Ein automatisierter oder teil-automatisierter Datenaustausch würde aber gegenüber dem Ist-Zustand auch die Bereitstellung deutlich größerer Datenmengen ermöglichen. Auf Grund der Menge von Emissionsdaten sollten überwiegend aggregierte Werte verwendet werden. Bei der

Veröffentlichung muss auf vertrauliche Daten geachtet werden. Diese können beim Export explizit gefiltert und als vertraulich markiert werden.

Die aktuellen Exporte (Excel) enthalten mehrere Aggregationslevel und sind über eine große Anzahl von Tabellenblättern verteilt. Die Struktur ist international vereinbart, um eine Vergleichbarkeit zu gewährleisten.

Von den Teilnehmenden kam die Einschätzung, dass der Organisationsaufwand zur Bereitstellung von Daten zur Umwelt derzeit größer ist als der technische Aufwand, die Daten in die entsprechenden Excel-Templates zu überführen. Dieser Vorgang geschieht aktuell zumeist manuell durch Kopieren der neuen Daten in die existierenden Excel-Dateien. Durch die aktuellen Templates müssen Daten nur auf einem Tabellenblatt ausgetauscht werden, die restlichen Blätter aktualisieren sich automatisch. Die Publikation von Daten zur Umwelt dauert oft sehr lange (bis zu Monaten), da das Schreiben von Texten sehr zeitaufwändig ist.

Für den Data Cube wäre es möglich, Baumstrukturen für Exporte zu definieren. Diese könnten entweder dateibasiert sein oder UBA intern auch über eine entsprechende API (durch Verwendung der Mesap DLL) direkt angebunden werden. Der aktuelle Austausch über Excel-Dateien könnte damit abgelöst werden. Eine genaue Definition der Anforderungen steht noch aus. Vor allem für Visualisierungen könnte das Data Cube Projekt auch für das FG V 1.6 interessant sein, da diese für Emissionen derzeit oft fehlen.

Durch die Teilnehmenden wurde weiterhin angemerkt, dass Drupal bereits einige Verwaltungstools bietet, deren Nutzbarkeit für den Data Cube von con terra analysiert werden sollten.

2.3.3 Fazit

Die bisherigen Gespräche mit datenhaltenden Stellen haben bestätigt, dass die Datengrundlage für den Data Cube sowohl technisch als auch fachlich sehr heterogen ist. Es gibt eine Vielzahl von Datenhaltungen/-formaten, welche in unterschiedlichste organisatorische Abläufe eingebunden sind. Bisher liefern die einzelnen datenhaltenden Stellen in verschiedenen zeitlichen Intervallen Daten, wofür eine Excel-Tabelle als Austauschformat verwendet wird. Unabhängig von den vorliegenden Systemen wird diese Tabelle überwiegend manuell gepflegt.

Vor allem bezogen auf den Umgang mit einzelnen Datenwerten (zum Beispiel dynamisches Umrechnen von Einheiten oder dynamische Berechnung von Aggregationen im Data Cube) wurde festgestellt, dass keine globalen Entscheidungen für das gesamte Projekt getroffen werden können. Das Vorgehen muss daher für jeden Datensatz beim Import definiert werden. Das Datenmodell sollte mit den verschiedensten Möglichkeiten umgehen können.

Zwischen den verschiedenen datenhaltenden Stellen und selbst innerhalb dieser existieren eine große Anzahl von verschiedenen Klassifikationen, welche zur Gruppierung und Aggregation von Daten verwendet werden. Diese sind jeweils kontextsensitiv und müssen selbst bei einer gleichen Bezeichnung nicht identisch sein. Es ist daher wichtig, für alle Datensätze Metainformationen über Klassifikationen abzulegen. Welche Datensätze durch den Data Cube verglichen werden können, muss von der Redaktion bzw. den jeweiligen Fachgebieten definiert werden und darf nicht durch die Nutzenden der Webseite frei gewählt werden.

Über alle Gespräche hinweg hat sich bestätigt, dass der organisatorische Aufwand zur Publikation von Daten derzeit sehr hoch ist. Bei allen datenhaltenden Stellen liegen weit mehr Daten vor als derzeit bereitgestellt werden. Es wurde bereits andiskutiert, ob in Zukunft auch mehr Daten ohne einen dazugehörigen Artikel veröffentlicht werden sollten.

Insgesamt wäre es für viele Teilnehmenden der Gespräche denkbar, die bisherige Excel-Tabelle als Austauschformat abzulösen, wenn damit die Effizienz und/oder Qualität der Datenlieferungen steigen würde.

2.4 Data-Store

Unter dem Begriff Data-Store wird die datenhaltende Komponente des Data Cubes verstanden, welche in der Lage sein soll, alle Daten zur Umwelt möglichst ohne Redundanzen zu speichern. Hierzu muss ein geeignetes Datenbank-Managementsystem (DBMS) ausgewählt und mit entsprechendem Datenmodell konzipiert werden. Zum Thema Data-Store wurde kein gesonderter Workshop abgehalten. Die Anforderungen leiten sich aus der Leistungsbeschreibung (Umweltbundesamt, 2020 - a), den Aussagen der Teilnehmenden der Workshops zu den Anforderungsanalysen und der Fachgespräche mit den datenhaltenden Stellen (siehe Kapitel 2.3) ab. Im Folgenden werden die Anforderungen an die Datenhaltung und -verarbeitung formuliert.

- Datenhaltung
  - Es ist eine zentrale Datenhaltung mit allen Daten zur Umwelt aufzubauen.
  - Die Datenstrukturen müssen klar definiert werden. Es muss ein themenübergreifender Gesamtansatz für das Datenmodell ausgearbeitet werden, in den sich die verschieden-artigen Daten zur Umwelt einordnen.
  - Das Datenmodell muss so konzipiert werden, dass es einerseits beliebig vertieft/ detailliert und andererseits beliebig fachlich erweitert werden kann.
- Datenmanagement
  - Die datenhaltenden Stellen müssen automatisiert ihre Daten im Data-Store einspielen können.
  - Auch rückwirkend müssen Datenaktualisierungen und -erfassungen im Data-Store an die Fachsysteme (teil-)automatisiert übergeben werden können.
  - Berichte können ihre Daten direkt aus dem Data-Store beziehen.
  - Die Webseite kann Daten direkt aus dem Data-Store nutzen.
  - Datenänderungen sollen automatisch an die Redakteure bzw. Fachexperten/-innen übermittelt werden.
  - Die Daten müssen über Status so markiert sein, dass sofort erkennbar ist, in welchem Bearbeitungszustand sie sich gerade befinden.
  - Dynamische Zugriffe auf die Daten müssen in Abhängigkeit des Status definiert und kontrolliert werden können.
- Dimensionen
  - Es muss möglich sein, Daten aus verschiedenen Quellsystemen (zum Beispiel DESTATIS, aber auch vielen weiteren Quellen) zu übernehmen, diese einheitlich strukturiert zu verwalten, sodass sie dann als Dimensionen der unterschiedlichen Daten zur Umwelt im Data-Store genutzt werden können.

- Es muss möglich sein, für alle erfassten Datensätze Dimensionen zu definieren.
- Es muss möglich sein, die übernommenen Daten (konkrete Werte der Datensätze) den entsprechenden Dimensionen zuzuordnen.

► Werte und Berechnungen

- Konkrete Daten müssen als Werte (Zahlen, Texte) gespeichert werden, sodass mit ihnen gerechnet werden kann.
- Es soll möglich sein, im Data-Store Berechnungen zu hinterlegen, die automatisch ausgeführt werden. Diese Anforderung kann ggf. während des Data-Inputs umgesetzt werden.
- Es soll möglich sein, bestimmte Berechnungen manuell anzustoßen.
- Es soll möglich sein, Berechnungen hierarchisch auszuführen. Das gilt auch für automatisierte Berechnungen.
- Es muss auch weiterhin möglich sein, extern mit den Daten zur Umwelt zu arbeiten und externe Daten mit ihren Bezügen zu den Ausgangsdaten im Data-Store einzuspielen.

► Historisierung

- Die Werte müssen historisch verwaltet werden.
- Bei hierarchischen Berechnungen müssen die Bezüge der Daten jederzeit wieder selektierbar sein.
- Objektklassen, die für die Dimensionierung herangezogen werden, müssen ebenfalls historisch verwaltet werden, um ältere Daten zur Umwelt auch weiterhin einordnen zu können.

2.5 Data-Input

Unter der Komponente Data-Input wird der Import verschiedener Datensätze der datenhaltenden Stellen in den Data-Store beschrieben. Für den Data-Input wurde kein gesonderter Workshop abgehalten. Die folgenden Anforderungen ergeben sich daher vor allem aus den Gesprächen mit den datenhaltenden Stellen (siehe Kapitel 2.3) und den Anforderungen an die verbleibenden Komponenten des Data Cube.

Durch den Data-Input müssen aus verschiedenen Datenquellen die Daten, die in unterschiedlichsten Formaten vorliegen können, in den Data-Store importiert werden können. Da der Data-Store zum aktuellen Zeitpunkt noch nicht final definiert ist, muss die Zielstruktur an dieser Stelle noch offengehalten werden. Wie in Kapitel 2.3 beschrieben, liegen die Quelldaten sowohl in verschiedenen Datenbanken bzw. Data-Warehouses dateibasiert (zum Beispiel Excel) sowie als Web-Services vor. Da die datenhaltenden Stellen verschiedene fachliche Themen bearbeiten, existieren auch inhaltlich viele verschiedene Datenstrukturen. Der Data-Input soll Daten aus den verschiedenen Stellen lesen und importieren können. Die Komponente muss daher die verschiedenen Formate unterstützen und die inhaltliche Struktur für den Data-Store anpassen können. Dieser Schritt wird als Schema-Transformation beschrieben und muss entsprechend des Datenmodells im Data-Store durchgeführt werden, falls eine strukturierte Datenhaltung (Datenbank, Data-Warehouse) und kein Data-Lake verwendet wird. Wesentliche Arbeitsschritte, welche für die meisten Datenbankmodelle notwendig sind, sind zum Beispiel die

Umbenennung von Attributen, Aufsplitten von verschiedenen Aggregationsebenen, Zuordnung von Dimensionen und Wertelisten, sowie die Identifikation und Neustrukturierung von Metadaten.

Zusätzlich zu den konkreten Daten müssen auch Metadaten in den Data-Store übernommen werden. In diesem Kontext sind Metadaten Zusatzinformationen, wie zum Beispiel Quelle, Autor, Datum, aber auch inhaltliche Angaben zur Klassifikation von Attributen (vgl. Kapitel 3.1.1.3). Sollten die Metadaten nicht maschinenlesbar in der Quelle vorliegen, sollen die Informationen manuell im Data-Input angegeben werden können. Durch diese manuelle Methode sollen Datensätze auch mit zusätzlichen Metadaten angereichert werden können. Für den Data-Input sollen im besten Fall die originalen Datenquellen verwendet werden, um keinen Mehraufwand zur Erstellung von Zwischenformaten zu erzeugen. Sollte dies nicht möglich sein, soll auf die derzeit verwendete Excel-Tabelle zurückgegriffen werden können, um Daten an den Data Cube zu übermitteln.

Im Data-Input sollen Datensätze auch untereinander verlinkt werden können, um später verwandte Datensätze auffindbar zu machen. Verwandte Datensätze können beispielsweise Datensätze zu einem gleichen Thema mit unterschiedlichen Einheiten oder Aggregationsebenen sein.

Mit dem Data-Input soll es weiterhin möglich sein, verschiedene Versionen eines Datensatzes zu importieren.

Für Datensätze mit gleichbleibenden Strukturen soll ein automatisierter Importprozess möglich sein. Das bedeutet, dass der Data-Input-Prozess für den jeweiligen Datensatz nicht für jeden Import angepasst werden muss und bei Bedarf direkt ausgeführt werden kann. Für Automatisierungen soll es zusätzlich möglich sein, zeitliche Ablaufpläne zu konfigurieren, um Datensätze regelmäßig zu importieren. Dieser Ansatz könnte zum Beispiel bei fest definierten Datenstrukturen wie Web-Services oder Datenbanken verwendet werden.

Für dateibasierte Datensätze soll eine Nutzeroberfläche verwendet werden können, um neue Datensätze hochzuladen und so für den Data Cube bereitzustellen. Eine Nutzeroberfläche muss hierzu die Möglichkeit bieten, verschiedene Pflichtangaben, wie zum Beispiel den Titel des Datensatzes oder Angaben zur Versionierung, entgegenzunehmen.

Im Data-Input sollen Schritte zur Qualitätssicherung (QS) durchgeführt werden können. Diese müssen je Datensatz individuell definiert werden. Mögliche QS Schritte sind zum Beispiel die Validierung von Wertebereichen (zum Beispiel ob ein Temperatur-Wert in einer vordefinierten Spanne liegt), der Überprüfung auf Vollständigkeit (zum Beispiel ob sämtliche Pflicht-Attribute vorhanden und gefüllt sind) oder auch die Identifikation von doppelten Datensätzen.

2.6 Technische Rahmenbedingungen durch das UBA

Im folgenden Kapitel werden technische Rahmenbedingungen durch das UBA besprochen, welche bei der Konzeption von Data Cube Lösungskomponenten beachtet werden müssen.

Für die Rahmenbedingungen im UBA Rechenzentrum wurde kein gesonderter Workshop abgehalten. Es wurde sich jedoch im Projektverlauf darauf geeinigt, verschiedene bereits existierende Richtlinien für die Konzeption zu verwenden. Als wesentliche Grundlage wird die Architekturrichtlinie für die IT des Bundes (Der Beauftragte der Bundesregierung für Informationstechnik, 2020) , sowie die Leitlinie für Informationssicherheit in der Bundesregierung (Bundesministerium des Innern, für Bau und Heimat, 2017) genutzt. Die darin beschriebenen Anforderungen müssen von allen Komponenten des Data Cube beachtet werden.

Darüber hinaus wurden die folgenden UBA spezifischen Dokumente zur Beachtung bereitgestellt:

- UBA Leitfaden für Webanwendungen
- Anforderungen an externe Webseiten oder Fachanwendungen des UBA im Web
- Corporate Design des UBA
- Online Styleguide des UBA

Für die Integration der Data-Outputs in das Drupal CMS des UBA fand ein Gespräch mit Teilnehmer von werk21 GmbH, sowie mit Teilnehmern des UBA von PB 2 statt. Die wesentlichen Erkenntnisse des Termins sind, dass für die Integration keine iFrame Lösungen verwendet werden können, da diese nicht barrierefrei und responsiv implementierbar sind. Das hat zur Folge, dass für die Einbettung eigene Drupal-Module implementiert werden müssen, wodurch nur Komponenten mit einer Integration-API oder Eigenentwicklungen für den Data-Output in Frage kommen. Komplexere Anwendungen (z. B. Dashboards) können grundsätzlich verlinkt werden, falls eine Integration in Drupal aus technischer Sicht nicht sinnvoll ist. Eine native Integration ist jedoch vorzuziehen. Dabei ist zu beachten, dass nur Inhalte (z. B. JavaScript Bibliotheken) geladen werden dürfen, die sich lokal auf dem Server des UBA befinden. CDNs können nicht verwendet werden. Für Data-Outputs wurde zusätzlich definiert, dass auch die aus dem UBA Styleguide definierten Farben für interaktive Diagramme verwendet werden sollen. Falls die verfügbaren Farben nicht ausreichend sind, könnten Anpassungen mit PB 2 abgestimmt werden.

Darüber hinaus wurde die Befüllung des Drupal-Suchindex besprochen. Eine Befüllung durch externe Anwendungen (z. B. ETL Prozesse) ist nicht möglich. Die Drupal-Suche kann daher nur für Drupal-Modul-Inhalte verwendet werden. Üblicherweise werden Schlüsselwörter bei der Erstellung von Drupal-Inhalten (z. B. Drupal-Artikel, Easycharts-Inhalten) definiert, die für die Suche verwendet werden. Eigene Drupal-Module müssten auf die gleiche Methode in die Suche eingebunden werden.

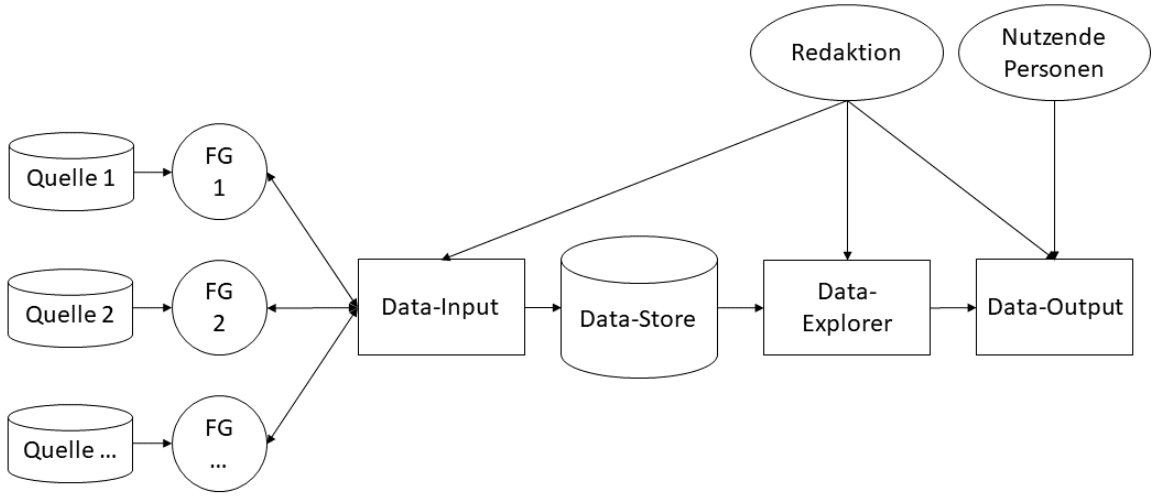
Bezüglich des Nutzermanagements wurde festgestellt, dass nicht alle zukünftigen Nutzenden des Data Cube einen Account im CMS besitzen. Ebenso kann kein single-sign-on Mechanismus aus dem UBA für eigene Entwicklungen angebunden werden. Für den Data Cube ist daher ein eigenes Nutzermanagement aufzubauen.

3 Konzeption

In diesem Kapitel werden erste Erkenntnisse und Ideen aus den Anforderungsworkshops skizziert. Da die Anforderungsanalyse noch nicht abgeschlossen ist, sind die einzelnen Kapitel ausdrücklich als Arbeitsstand und nicht als vollständige Konzeption zu betrachten.

Um die einzelnen Komponenten in Relation zu betrachten, zeigt Abbildung 1 die grundlegende Architektur. Von links nach rechts wird dabei gleichzeitig der Datenfluss beschrieben. Von den einzelnen datenhaltenden Stellen werden verschiedenen Quellen über die Data-Input Komponente durch die einzelnen Fachgebiete oder die Redaktion in den Data-Store überführt. Anschließend können die Daten im Data-Explorer durch die Redaktion eingesehen und zu Data-Outputs zusammengestellt werden. Die finalen Data-Outputs werden veröffentlicht und den Nutzenden der UBA-Webseite zur Verfügung gestellt.

Abbildung 1: Vereinfachte Architektur der Data Cube Komponenten



Quelle: eigene Darstellung, con terra GmbH

3.1 Data-Store

Im Folgenden werden erste Ansätze für die Datenverwaltung und das Datenmanagement aufgestellt. Die konkreten Anforderungen an die Komponente Data-Store sind in Kapitel 2.4 thematisiert. In diesem Kapitel geht es primär um das Speichern der Daten und nicht um Bedienungen, Datennutzungen oder Mensch-Maschine-Schnittstellen. Die im Folgenden beschriebenen Ansätze sind in der weiteren Konzeptionsphase des Projektverlaufs zu diskutieren, zu präzisieren und zu spezifizieren.

3.1.1 Technische Voraussetzungen und Begriffserklärungen

Um die Komponente Data-Store beschreiben zu können, werden zunächst einige technische Voraussetzungen und Begriffe erklärt, bevor in Kapitel 3.1.4 konkret auf Lösungsansätze eingegangen wird.

3.1.1.1 Objekte, Objektklassen und Klassendiagramme

Mit der Digitalisierung wird eine zweite, virtuelle Welt erschaffen. Für jeden digital zu erfassenden Gegenstand oder Prozess müssen zunächst digitale Strukturen definiert werden. Diese Aufgabe ist mittlerweile recht komplex, was sich einerseits mit den digital abzubildenden, oft

hoch komplexen Themen erklärt und andererseits mit der Erfassung von Verknüpfungen, Wechselwirkungen oder Hierarchien einhergeht. Um diese abstrakte Welt besser beschreiben zu können, werden zunächst einige gängige Konzepte kurz erläutert. In diesem Kontext ist zu beachten, dass in einem Computer nur das ausgewertet und analysiert werden kann, was in den digitalen Daten strukturell angelegt ist.

Als Objekte werden Gegenstände, Prozesse und Sachverhalte der Realität bezeichnet, die digital abgebildet werden. Objekte werden zu Klassen zusammengefasst, wenn sie digital auf dieselbe Art und Weise abgebildet und strukturiert werden. Eine Objektklasse ist demzufolge ein Abbild einer Menge von Objekten, die eine gemeinsame Struktur und ein gemeinsames Verhalten aufweisen; eine Objektklasse ist eine strukturierte Zusammenfassung von Daten, ihren Eigenschaften, sogenannten Attributen, und Funktionen, sogenannten Operationen (vgl. Anhang 1 in (Rudolf, Umweltdatenmanagement – Eine Geo-Inspiration, 2018)). Im Kontext des Data Cube wäre eine Objektklasse zum Beispiel eine Excel-Tabelle mit einer bestimmten Formatvorlage, zu der eine genaue Definition der existierenden Tabellenblätter, der verwendeten Zeilen und Spalten sowie möglicher Rechenoperationen vorliegt. Einzelne Excel-Tabellen, welche dieser Vorlage entsprechen, sind in diesem Beispiel Objekte dieser Objektklasse.

Um komplexe Systeme aus Objektklassen und ihren Verknüpfungen zu beschreiben, werden sogenannte Klassendiagramme entwickelt. Zur Dokumentation wurden UML-Klassendiagramme als Standard etabliert (Object Management Group, 2005). Die Modellierungssprache UML legt fest, mit welchen Begriffen und Beziehungen die Zusammenhänge in einem Modell spezifiziert werden. Weiterhin werden mit der UML grafische Notationen in Diagrammen definiert (vgl. Anhang 1 in (Rudolf, Umweltdatenmanagement – Eine Geo-Inspiration, 2018)).

3.1.1.2 Dimensionen, Werte und Würfel (Cubes)

Zusätzlich zu den Begrifflichkeiten der Datenmodellierung werden vor allem zur Analyse von Daten häufig auch die Begriffe Dimensionen und Werte verwendet. Dimensionen beschreiben Kategorien oder Interessensgebiete, in denen Daten vorliegen. Werte beschreiben die konkreten Inhalte. Die Begriffe sollen anhand der Abbildung 2 an einem praktischen Beispiel erläutert werden. Die Abbildung zeigt eine Tabelle mit Werten zu CO2-Emissionen. Eine Zelle ist als konkreter Wert rot umrahmt. Die blau gekennzeichneten Zellen sind die Dimensionen. Der konkret ausgewählte Wert besitzt also die Dimensionen: „Manufacturing Industries and Construction (1.A.2)“ in der Zeile, das Jahr 2005 in der Spalte und den Stoff CO2 im ausgewählten Tabellenblatt.

Abbildung 2: Dimensionen und Werte anhand eines Beispiels

Emission trends for Germany since 1990, CO_2 in kt

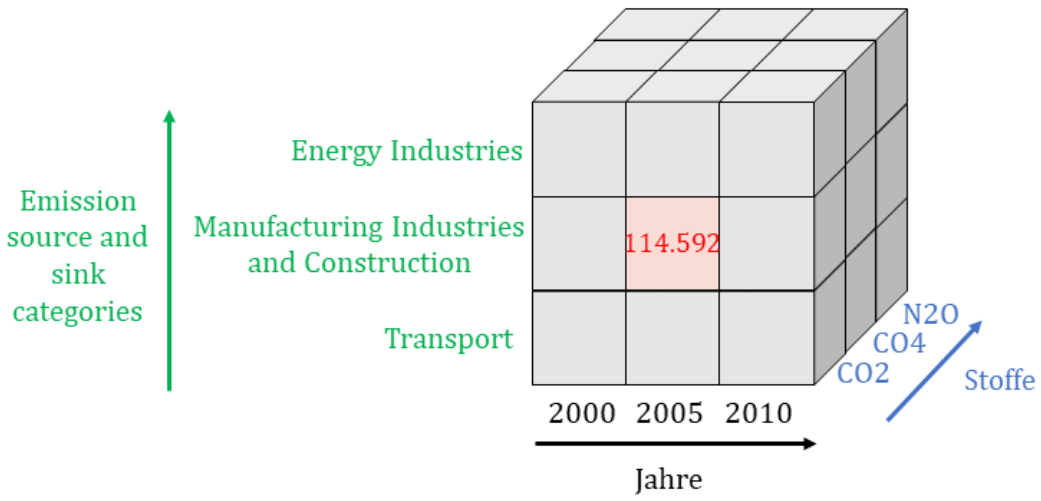
Emission source and sink categories	1990	1995	2000	2005	2010	2011	2012
Total Emissions (without LULUCF)	1.052.477	938.968	899.852	866.697	832.949	809.217	813.985
Total Emissions (with LULUCF)	1.074.783	902.697	876.761	863.715	820.112	797.191	793.061
1. Energy	989.590	881.052	839.701	812.072	784.476	760.607	766.053
A. Fuel Combustion	985.750	877.992	836.709	809.118	781.894	757.935	763.396
1. Energy Industries	423.906	364.609	355.168	375.878	351.642	349.074	358.561
2. Manufacturing Industries and Construction	185.165	144.546	129.219	114.592	124.749	121.826	116.976
3. Transport	161.927	175.094	180.604	160.314	153.040	154.882	153.543
thereof Road transportation	*151.886*	*166.451*	*172.541*	*153.040*	*146.752*	*148.677*	*147.355*
4. Other Sectors	202.954	189.739	169.387	156.616	151.154	130.939	133.313
thereof Commercial / Institutional	*64.111*	*53.108*	*45.512*	*40.041*	*39.998*	*35.896*	*34.379*
thereof Residential	*128.636*	*128.973*	*117.780*	*110.967*	*105.502*	*89.553*	*93.712*
5. Other (military)	11.797	4.005	2.331	1.718	1.309	1.214	1.002
B. Fugitive Emissions from Fuels	3.840	3.060	2.992	2.953	2.581	2.673	2.657
1. Solid Fuels	1.833	933	779	741	684	683	688
2. Oil and Natural Gas	2.008	2.127	2.213	2.212	1.898	1.990	1.969
2. Industry	59.695	55.788	57.496	52.248	45.956	46.098	45.296
A. Mineral Industry	23.522	24.487	23.266	20.126	18.952	20.151	19.666
B. Chemical Industry	8.109	7.966	8.443	8.748	8.297	8.074	8.223
C. Metal Industry	25.080	20.794	23.460	21.138	16.399	15.693	15.240
D. Non-Energy Products from Fuels	2.983	2.541	2.327	2.237	2.308	2.179	2.167
E. Electronics Industry							

Deckblatt_Cover | Anmerkungen_Notes | Trend_Summary | GHG_Summary | GHG_Fractions | GHG_CO2eq | CO2 | CH4_CO2eq | CH4 | N2O_CO2eq | N2O | F-Gases_CO2eq | HFC_CO2eq | P ...

Quelle: Umweltbundesamt – Daten zur Umwelt, markiert von hrd.consulting

Zur Veranschaulichung werden Daten mit mehreren Dimensionen häufig als Würfel (Cubes) beschrieben. Dabei sind die Dimensionen die einzelnen Achsen des Würfels. Datensätze können im Allgemeinen mehr als drei Dimensionen haben, allerdings sei dies zur Begriffserklärung vernachlässigt, um die Darstellung zu vereinfachen. Abbildung 3 zeigt die identifizierten Dimensionen der Excel-Tabelle schematisch als farblich markierte Achsen eines Würfels. Der rot markierte Wert in der Excel-Tabelle in Abbildung 2 ist ebenfalls im Zentrum der vorderen Ansicht des Würfels eingetragen. Diese Position entspricht genau den Ausprägungen der zugehörigen Dimensionen.

Abbildung 3: Dimensionen veranschaulicht am Würfel-Modell



Quelle: eigene Darstellung, con terra GmbH

3.1.1.3 Metadaten

Metadaten sind Informationen über Daten. Sie können ebenfalls wie die ursprünglichen Daten verwaltet und verarbeitet werden. Dabei können zwischen semantischen und technischen Metadaten unterschieden werden. Semantische Metadaten „[…] beinhalten Informationen über den Inhalt, räumlich-zeitliche Bezüge, die Datenqualität, Zugangsmöglichkeiten oder Nutzungsrechte und beschreiben damit die Eignung von Daten für bestimmte Anwendungszwecke, Präsentations- und Verarbeitungsmethoden“ (vgl. Anhang 1 in (Rudolf, Umweltdatenmanagement – Eine Geo-Inspiration, 2018)). Technische Metadaten hingegen „[…] werden oft als ‚Ontologien‘ bezeichnet. Sie beschreiben die Struktur von Datensammlungen und Regeln der Datenverarbeitung, zum Beispiel Definitionen der Objekte, ihrer Verknüpfungen, der einzelnen Datenfelder und ihrer Verarbeitungskonventionen“ (vgl. Anhang 1 in (Rudolf, Umweltdatenmanagement – Eine Geo-Inspiration, 2018)). Der Begriff wird daher häufig je nach Kontext unterschiedlich verwendet.

Anknüpfend an das vorherige Beispiel einer Excel-Tabelle, die einer bestimmten Formatvorlage unterliegt, nennen wir exemplarisch als mögliche semantische Metadaten Eigenschaften wie den Autor der Tabelle, den Ablageort einer physischen Kopie sowie das Erstellungsdatum oder das letzte Aktualisierungsdatum.

3.1.1.4 Interoperabilität

Die digitale Transformation setzt zwingend voraus, dass digitale Daten interoperabel sind, das heißt, über verschiedene Anwendungen hinweg weiterverarbeitet werden können. Dies bedeutet, dass einerseits die technische Übertragung via Schnittstellen (zum Beispiel JDBC für relationale Datenbanken), via standardisierter Web-Dienste (zum Beispiel WMS, WFS, REST, REST-API), oder über diverse Austauschformate unterstützt werden muss. Andererseits bedeutet das aber auch, dass die Dateninhalte und -strukturen vereinheitlicht sein müssen, um sie auswerten zu können.

In den letzten Jahren wurden immer mehr datenstrukturelle Standards für die Datenbereitstellung ausgearbeitet (zum Beispiel SDMX, GML, X-ÖV, X-PLANUNG, OKSTRA, INSPIRE). Sie definieren die Datenmodelle in Form von UML-Klassendiagrammen, die auf den ISO-Normen der ISO 191xx-Serie aufsetzen.

Am Beispiel der Excel-Tabelle wird hervorgehoben, dass Interoperabilität bedeutet, dass Schnittstellen (APIs) existieren müssen, die das Auslesen oder Setzen spezieller Daten in der Tabelle allgemeingültig für verschiedene Anwendungen oder Benutzende erlauben.

Nachdem erste technische Begrifflichkeiten erläutert wurden, wird im Folgenden das Datenmanagement genauer beleuchtet. Dazu werden die Konzepte einer Datenbank und eines Data-Lakes betrachtet.

3.1.2 Vergleich von Datenbank und Data-Lake

Aus den obigen Zusammenhängen lassen sich die zwei Hauptfunktionen Datenhaltung und Datenbereitstellung für den Data-Store ableiten. Im Folgenden werden zunächst wesentliche Konzepte erläutert und anschließend mit Bezug auf das Data Cube Projekt bewertet.

3.1.2.1 Konzepte

Eine zentrale Fragestellung zur Konzeption des Data-Store ist die Wahl der Datenhaltung, wobei hier grundsätzlich zwischen strukturierten und unstrukturierten Methoden unterschieden wird. Während es bei der Datenhaltung jedoch primär darum geht, wie Daten vorgehalten werden

sollen, muss für den Data Cube auch die darauffolgende Bereitstellung der Daten für die Komponenten Data-Explorer und Data-Output betrachtet werden.

Unstrukturierte Datenhaltungen werden häufig als Data Lakes („Daten See“) (Wikipedia, o. J.) bezeichnet. In diesen werden vor allem dateibasierte Daten in ihrer Rohform gespeichert. Daten werden in der Regel so verwaltet, wie sie entstanden sind, wobei keine festen technischen Vorgaben zur internen Datenstruktur existieren. Data Lakes sind Datenablagen ohne strukturelle Zusammenhänge. Das können sowohl unstrukturierte Dateien (zum Beispiel PDF, Word, etc.), aber auch in sich strukturierte Einzeldateien (zum Beispiel Excel, CSV, JSON) sein. Die Dateien stehen jedoch in keinem Kontext mit anderen Daten. Data Lakes können in verschiedenen Arten verwaltet werden. Möglichkeiten sind unter anderem als Dateiverzeichnis, oder innerhalb einer Datenbank.

Data Lakes sind sehr gut zum Speichern von heterogenen Datenquellen geeignet, da Dateien ohne Anpassung gespeichert werden und somit kein großer Aufwand beim Import entsteht. Dieser Ansatz wird in der Regel gewählt, wenn mehrere verschiedene Anwendungsfälle für die Quelldaten existieren, oder im Gegenteil noch kein Anwendungsfall bekannt ist. Durch die Speicherung der Daten in Rohform gehen keine Informationen beim Speichern verloren.

Der Nachteil eines Data Lake ist, dass die Auswertung der Daten auf die jeweilige Zielanwendung verlagert wird. Je nach Anwendungsfall müssen Daten aus verschiedenen Quellen zusammengeführt und strukturiert werden. Data Lakes werden daher vor allem als initialer Speicherort verwendet, bis ein geeigneter Prozess zur Strukturierung etabliert ist.

Im direkten Vergleich hierzu stehen strukturierte Datenhaltungen. Hierzu wird häufig eine relationale Datenbank verwendet. Innerhalb einer Datenbank werden Daten strukturiert in einzelnen Tabellen abgelegt, wobei auch zwischen den einzelnen Tabellen Verknüpfungen definiert sein können. Daten werden so verwaltet, dass jede Objektklasse beschrieben wird und ihre Verknüpfungen definiert sind. Einer Datenbank liegt fast immer ein Klassendiagramm zugrunde, dass die Datenstrukturen der Objektklassen, sowie ihre Verbindungen zueinander definiert und dokumentiert. Für eine Datenbank müssen Daten vor dem Import eingelesen und entsprechend dem Zielschema der Tabelle aufbereitet werden. Das Zielschema beschreibt dabei alle Spalten (Attribute) und die dazugehörigen Daten-Typen (Texte, Zahlen, Binärdaten). Je nach Schema kann es vorkommen, dass nicht alle Informationen aus einer Datei übernommen werden können. Es kommt daher zu einem Informationsverlust. Während das Importieren von Daten in die Datenbank wesentlich aufwendiger als das Speichern von Daten im Data Lake ist, ist die Datenbereitstellung speziell für Anwendungen wesentlich einfacher. Der Vorteil einer strukturierten Datenhaltung ist, dass alle Strukturen bekannt sind und Anwendungen daher nur für wenige feste Strukturen implementiert werden müssen. Zusätzlich steigt die Geschwindigkeit der Datenbereitstellung, da das Einlesen der Daten bereits beim Import durchgeführt wurde.

Neben der Datenhaltung müssen die erfassten Daten so aufbereitet werden, dass sie für Anzeigen, Berechnungen, Auswertungen und Ähnliches zur Verfügung gestellt werden können (Datenbereitstellung). Dazu müssen im Data-Store entsprechende Zugriffsmöglichkeiten bereitgestellt werden. Das kann entweder über direkte Zugriffe auf die Datenhaltungskomponente (zum Beispiel via SQL auf eine Datenbank) oder über Dienste-Schnittstellen (zum Beispiel als Download-Services in WFS oder als API-Features-Dienst) erfolgen.

Grundsätzlich wird bezüglich des Datenflusses zwischen ETL (Extract Transform Load) und ELT (Extract Load Transform) unterschieden. Bei den Begriffen wird davon ausgegangen, dass die drei Schritte (Extrahieren von Dateien aus der Quelle, Transformieren der Daten in das Zielschema, und Speichern der Daten im Zielsystem) immer ausgeführt werden müssen, jedoch lediglich die

Reihenfolge austauschbar ist. ELT wird daher üblicherweise für Data Lakes, ETL für Datenbanken verwendet.

3.1.2.2 Fazit

Der Betrieb einer Anwendung (Data-Explorer, Data-Output) ist nur sehr schwierig für unstrukturierte Daten (Data Lakes) umzusetzen, da die jeweilige Anwendung die Strukturierung der verschiedenen Quellen zur Laufzeit übernehmen müsste. Das bedeutet, dass zum Beispiel zur Anzeige eines interaktiven Diagramms Daten aus potenziell mehreren Quellen eingelesen werden müssten, sobald die entsprechende Webseite aufgerufen wird. Dies ist zum einen nicht sehr performant lösbar, zum anderen wird die Komplexität der Data-Explorer und Data-Output Komponenten stark erhöht, da Wissen über die verschiedensten Quellstrukturen innerhalb der Komponenten notwendig ist. Durch eine strukturierte Datenhaltung hingegen muss die Zielanwendung nur für einige wenigen Strukturen implementiert werden. Der ETL Prozess für den Data-Input kann separat von den Zielanwendungen implementiert werden, wodurch eine saubere Funktionstrennung möglich ist. Durch klare Strukturen können auch die Funktionalitäten der Anwendungen (zum Beispiel Abfragen, Filter, Aggregationen über verschiedene Datensätze) einheitlich über alle Datensätze hinweg bereitgestellt werden. Zusätzlich wird durch vorprozessierte Daten die Performance erhöht, da der Arbeitsschritt bereits vor der Datenabfrage auf der Datenbank durchgeführt wurde (ETL). Für die Data-Store Komponente wird daher eine strukturierte Datenhaltung in Form einer relationalen Datenbank empfohlen und im nächsten Schritt werden dazu verschiedene Datenmodelle näher thematisiert.

Abschließend wird ergänzt, dass es möglich ist, zusätzlich einen Data Lake einzubinden, auf den dann von der Datenbank aus verwiesen wird. Dieser Ansatz kann verwendet werden, um Zusatzdokumente wie zum Beispiel Bilder, Videos, PDF- oder Word-Dateien abzulegen, die nicht in eine strukturierte Form überführt werden können.

3.1.3 Vergleich von generischen und spezifischen Datenmodellen

Innerhalb von relationalen Datenbanken können zwischen generischen und spezifischen Datenmodellen unterschieden werden. Im Folgenden werden die Unterschiede beschrieben.

3.1.3.1 Konzepte

In der Praxis werden zumeist spezifische Datenmodelle verwendet. Das heißt, Daten werden in einem speziell für den fachlichen Inhalt entworfenem Datenmodell verwaltet. So entstehen fachlich zugeschnittene, sogenannte proprietäre Datenhaltungen. Der Vorteil ist, dass diese Datenmodelle effizient für individuelle Anwendungsfälle funktionieren. Da einzelne Tabellen und deren Verlinkungen für spezifische Anforderungen entworfen werden, können Quelldatensätze oft mit geringem Aufwand übernommen werden, da sich die Strukturen von Quelle und Ziel im besten Fall sehr ähnlich sind. Zusätzlich können Tabellen in der Datenbank direkt zur Datenbereitstellung verwendet werden. Der Nachteil ist jedoch, dass spezifische Systeme oft sehr unflexibel sind. In vielen Fällen führt dies dazu, dass für neue Anforderungen (zum Beispiel Anpassungen am Schema einer Datenlieferung) auch neue Tabellen in der Datenbank entworfen werden müssen, oder Informationen beim Import verloren gehen. Für den Data Cube bedeutet dies, dass für alle Datenstrukturen der verschiedenen datenhaltenden Stellen eigene Ziel-Tabellen angelegt werden müssten. Bei Änderungen an den Quellstrukturen müsste das bestehende Schema entweder angepasst werden, oder es müssten weitere Tabellen hinzugefügt werden.

Als Alternative zu den spezifischen Modellen existieren generische Datenmodelle, welche Daten in einer allgemeingültigen Form, unabhängig des Inhalts und der Quelle ablegen. Hierzu werden

Daten üblicherweise in ihre Einzelkomponenten (Dimensionen und Werte) aufgesplittet und separat abgespeichert. Diese Form der Datenhaltung benötigt technische Metadaten, die die Inhalte der generisch angelegten Objektklassen verwalten. Das heißt, Daten werden in Datentabellen mit allgemeingültigen Strukturen übernommen, die Dateninhalte und die konkreten strukturellen Ausprägungen (insbesondere die Verlinkungen) werden in zentralen Metadatentabellen abgelegt. Beim Import werden die (Fach-)Daten in die generische Form überführt und ggf. aufgesplittet. Dieser Schritt wird üblicherweise maschinell übernommen, da das Aufsplitten der Daten manuell aufwendig sein kann. Zur Datenbereitstellung können die Daten entsprechend wieder zusammengefügt werden. Dabei können für verschiedene Anwendungsfälle verschiedene Darstellungen gewählt werden. Das heißt, dass für Datensätze verschiedene Kombinationen aus Dimensionen erzeugt werden können. Dies kann durch vordefinierte Datenbank-Sichten, sogenannte Views, auf die entsprechenden Daten oder durch direkte SQL-Abfragen ermöglicht werden. Dadurch kann unter anderem ein spezifisches Modell auf Basis eines generischen Modells bereitgestellt werden. Der Vorteil hierbei ist weiterhin, dass durch Sichten verschiedene Perspektiven auf die gleichen Daten bereitgestellt werden können. Die Datenhaltung ist daher von der Art der Darstellung getrennt.

3.1.3.2 Fazit

Wie in Kapitel 2.3 beschrieben, muss der Data-Store Daten aus vielen verschiedenen Themengebieten speichern können. Dabei ist schon jetzt eine Vielzahl verschiedener Strukturen bekannt. Es ist weiterhin zu erwarten, dass sich Strukturen über die Zeit ändern werden und neue Datensätze mit aufgenommen werden müssen. Grundsätzlich ist sowohl ein spezifisches als auch ein generisches Modell für den Data-Store nutzbar. Während spezifische Datenmodelle initial einfacher zu verstehen und anzuwenden sind, muss das Modell konstant gepflegt und weiterentwickelt werden. Dies kann auch dazu führen, dass Anpassungen an den Data-Output oder Data-Explorer Komponenten vorgenommen werden müssen. Ein generisches Modell hingegen kann stetig neue Daten, die in DzU bereitgestellt bzw. übernommen werden sollen, aufnehmen, ohne dass Anpassungen am Modell notwendig sind. Dafür ist das Datenmodell jedoch abstrakt und benötigt zum Verständnis Metadaten. Das setzt zwangsläufig eine programmgesteuerte Datenverarbeitung voraus.

Je nach Wahl der Data-Output und Data-Explorer Komponente müssen in einem generischen Schema Sichten auf die Daten definiert werden, um die neuen Daten nutzbar zu machen. Die Definition von Sichten läuft für alle Strukturen gleich ab, die notwendigen Informationen dazu sind im Datenmodell selbst festgeschrieben. Dadurch können auch Schnittstellen entworfen werden, welche die entsprechenden Sichten eigenständig aufbauen und so nutzbar machen können.

Eine finale Entscheidung kann hier erst nach der weiteren Evaluation von konkreten Lösungen getroffen werden. Diese werden in Kapitel 3.1.5 während der Konzeptionsphase beschrieben.

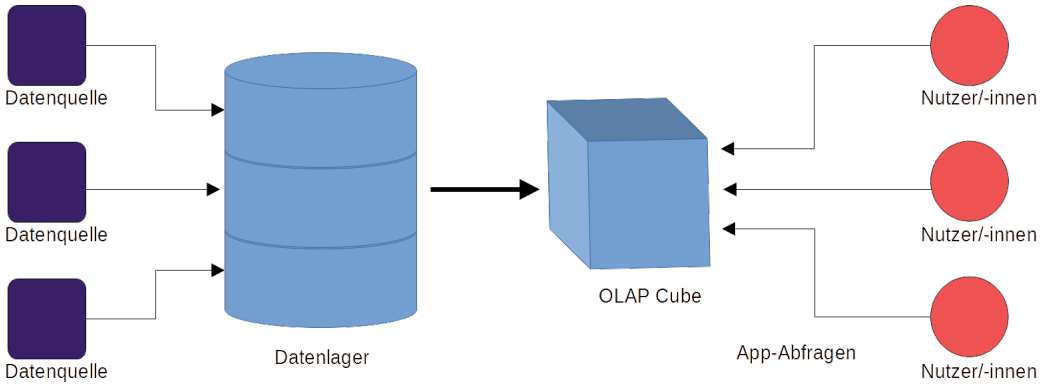
3.1.4 Datenbereitstellung

In den vorherigen Kapiteln wurde vor allem das Thema Datenhaltung erläutert, wobei die Dateibereitstellung nur kurz angerissen wurde. Grundsätzlich gilt es zu bedenken, dass die Bereitstellung nicht zwangsläufig durch die Datenhaltung vorgegeben sein muss, sondern durch zwei getrennte Komponenten abgebildet werden können.

Ein konkretes Beispiel wie Daten aus der Datenhaltung bereitgestellt werden können, wird durch den OLAP-Prozess beschrieben. Abbildung 4 zeigt das übliche Vorgehen der OLAP-Technologie („OnLine Analytical Processing“). Daten werden dabei zunächst in einer zentralen

Datenhaltung gespeichert und erst im nächsten Schritt als OLAP Cubes aufbereitet und entsprechend zur Verfügung gestellt.

Abbildung 4: Der OLAP-Prozess



Quelle: eigene Darstellung, hrd.consulting, in Anlehnung an den OLAP-Prozess in Abbildung 1 in (Redaktion ComputerWeekly.de, TechTarget, Inc., o. J.)

Die in Kapitel 3.1.1.2 dargestellte Würfel-Form ist angelehnt an das Konzept von OLAP-Cubes. Die OLAP-Technologie ist bekannt für gute Performance und hohe Flexibilität. Zudem ist es auf Basis von OLAP-Cubes möglich, bestimmte Transformationen, Filteroptionen, Aggregationen sowie Disaggregationen (zum Beispiel sogenannte Slice-, Dice-, Drill-Down-, Roll-up- oder Rotate-Operationen, vgl. (Redaktion ComputerWeekly.de, TechTarget, Inc., o. J.), (Luber, 2017)) zu realisieren, wofür häufig eine spezielle Abfragesprache (zum Beispiel MDX) verwendet wird (Luber, 2017). Dennoch ist diese Art der Datenverarbeitung und -vorhaltung auch limitiert und beschränkt, insbesondere im Kontext von heterogenen und komplexen Daten und Datenstrukturen (Mansmann, Ur Rehman, Weiler, & Scholl, 2014). Die Herausforderung der DzU an den Aufbau eines Data Cube besteht darin, dass jeder Datensatz durch seine eigenen Dimensionen beschrieben ist. Das heißt, die Dimensionen müssen flexibel und dynamisch ausgewählt und dem Datensatz zugeordnet werden können. Da die Dimensionen auch auf Basis von Daten externer Systeme (zum Beispiel aus DESTATIS) gebildet werden, ist eine einheitliche Verwaltung der infrage kommenden Dimensionen (auch der von Externen übernommenen Daten) in der Datenhaltung sinnvoll.

Die beschriebenen Würfel-Operationen (Roll-up, Drill-down, etc.) sollen durch die Anwendenden im Data-Explorer und Data-Output nutzbar sein. Hierbei sind jedoch verschiedene Ansätze denkbar, welche diese Funktionen bereitstellen könnten. Eine Alternative zu OLAP wäre zum Beispiel die Bereitstellung der Daten über eine eigene API, die von den Anwendungen verwendet werden würde. In der Darstellung (Abbildung 4), würde der OLAP Cube daher durch eine API-Komponente ersetzt werden. Die entsprechende Logik der Würfel-Funktionalitäten müssten in diesem Fall in durch die API abgedeckt werden. Eine weitere Möglichkeit ist die Generierung von verschiedenen Datenbank-Sichten auf Basis von SQL (siehe Kapitel 3.1.3). Innerhalb dieser können verschiedene Aggregationen und Filtermöglichkeiten für die Daten in der Datenhaltung bereitgestellt werde. Zuletzt könnten die genannten Funktionalitäten auch komplett durch die Anwendungen (Data-Explorer und Data-Output) bereitgestellt werden, wodurch keine eigene Komponente zur Datenbereitstellung notwendig wäre.

Im Rahmen der anstehenden Konzeptionsphase sollen unter anderem verschiedene externe Tools, insbesondere für die Komponenten Data-Explorer und Data-Output, verglichen werden. Gerade deren Anforderungen an die Datenhaltung und potenzielle Schnittstellendefinitionen für Datenbereitstellungen werden genauso in den Entscheidungsprozess mit einbezogen wie die Möglichkeit der Eigenentwicklung der einzelnen Teilkomponenten sowie der zugehörigen Kommunikationsschnittstellen. Konkret bedeutet dies, dass OLAP-Funktionalitäten nur dann sinnvoll verwendet werden können, wenn auch die Anwendungen die entsprechenden Abfragen (zum Beispiel durch MDX) durchführen können.

3.1.5 Lösungsalternativen

Im Verlauf der Konzeptionsphase werden verschiedene Lösungsalternativen betrachtet und mit Bezug auf die in Kapitel 2.4 beschriebenen Anforderungen evaluiert. Da der Data-Store als Schnittstelle zwischen Data-Input, Data-Explorer und Data-Output fungiert, müssen hier auch Kompatibilität und Auswirkungen auf die genannten Komponenten berücksichtigt werden.

Lösungen, die während der Konzeptionsphase betrachtet werden können, sind:

- Mesap
  - Die Software Mesap (siehe Kapitel 2.3.2.3) wird bereits im UBA zur Speicherung von Zeitreihen in Form von Bäumen eingesetzt.
- Generisches Datenmodell
  - Verwendung eines generisches Datenmodells (siehe Kapitel 3.1.3), welches mit einem allgemeingültigen Modell zur Speicherung aller Daten verwendet werden könnte (zum Beispiel envVisio (Rudolf, envVisio mit neuen Ansätzen im Umweltdatenmanagement: modelltheoretisch hergeleitet, fachlich ausgearbeitet, praktisch umgesetzt, 2020)).
- Spezifisches Datenmodell
  - Verwendung eines spezifischen Datenmodells (siehe Kapitel 3.1.3), welches auf Basis der Input-Daten speziell für den Data Cube entwickelt wird.
- Statistical Data and Metadata eXchange (SDMX) (Statistical Data and Metadata eXchange - SDMX Community, o. J. - f)
  - SDMX ist ein internationales Datenaustauschformat für statistische Daten aus verschiedenen Domänen, welches von verschiedenen Organisationen (zum Beispiel EUROSTAT und OECD) verwendet wird.

3.2 Data-Input

Die Daten, die in den Data Store übernommen werden sollen, zeichnen sich vor allem durch eine große Heterogenität aus. Die Daten unterscheiden sich sowohl inhaltlich als auch strukturell stark voneinander. Um die Daten in den Datenbestand übernehmen zu können, sind daher effiziente Methoden zur Datenintegration zu entwickeln. ETL (Extract Transform Load) Software eignet sich, um Daten in eine einheitliche Struktur und ein einheitliches Format zu konvertieren. Dabei werden die Daten zunächst aus dem Originalformat übernommen (Extract), dann umgewandelt (Transform) und abschließend in die Zielstruktur überführt (Load). Eine besondere Bedeutung hat dabei der Teil „Transform“. Mit diesem können eine Vielzahl von Bearbeitungsfunktionen in den Prozess integriert werden, wie zum Beispiel Formatumwandlung, Datenmodelltransformation oder Qualitätsprüfungen. ETL-Prozesse sind

automatisierbar und wiederholbar. Ein weiterer Vorteil ist die Möglichkeit, diese über Schnittstellen einer Server- oder Web-Applikation bereitzustellen. In der Projektvision führt dies dazu, dass die datenhaltenden Stellen zukünftig selbstverantwortlich aktualisierte Datensätze zur Verfügung stellen können.

Die folgenden Punkte müssen von den einzelnen ETL-Prozessen umgesetzt werden können:

- Einlesen der Quelldaten
- Einlesen von Metadaten
- Anreicherung von zusätzlichen Metadaten
  - Beschreibungen
  - Copyright
  - Berechnungsgrundlagen
  - Quellen
  - Name des Datensatzes
  - Informationen zur Versionierung
  - Bearbeiter
  - …
- Speichern der Daten im Data-Store
  - Es ist voraussichtlich eine Anpassung der Datenstruktur von Quelle zu Ziel notwendig.
- Verlinkung des Datensatzes mit anderen Daten im Data-Store
  - Verlinkung von verwandten Datensätzen
  - Verlinkung von Aggregationen / Disaggregationen
  - Verlinkung von Umrechnungen
- Überschreiben / Löschen bestehender Datensätze

Um eine einheitliche und effiziente Gestaltung der Prozesse zu ermöglichen, sollen Templates für ETL-Prozesse bereitgestellt werden. Diese können für alle grundlegenden Funktionen vorbereitet werden, um die Datenmigration durchzuführen. Für jeden Datensatz müssen Quelle (Format, Pfad, Datenstruktur) sowie das Ziel (Name des Datensatzes im Data-Store) jedoch individuell angegeben werden. Das bedeutet, dass für jeden Datensatz ein eigener ETL-Prozess, basierend auf dem Template, erzeugt wird.

3.2.1 Lieferung neuer Daten

Bei der Lieferung von neuen Daten müssen verschiedene Faktoren berücksichtigt werden.

Sollten neue Daten bei gleicher Struktur geliefert werden, kann der existierende Prozess erneut mit der neuen Datenquelle ausgeführt werden, um die Einträge in der Datenbank zu erweitern. Dieser Ansatz kann zum Beispiel für Zeitreihen verwendet werden, bei denen jeweils nur die neusten Informationen zu den bestehenden Zeitreihen hinzugefügt werden sollen (zum Beispiel

Neue Temperatur-Messungen mit neuen Zeitstempeln für eine Messstelle). Hierzu ist es notwendig, dass die einzelnen Einträge in der entsprechenden Tabelle eindeutig identifiziert werden können (ID, Zeitstempel, oder Kombination aus mehreren Attributen), um doppelte Einträge zu vermeiden.

Hat sich die Struktur der Quelldaten geändert, muss der ETL-Prozess für die neue Struktur angepasst werden. Beispiele sind Anpassungen von Attribut-Namen, welche neu zugeordnet werden müssen, oder auch die Selektion von Zellen mit Werten einer Excel-Tabelle.

3.2.2 Lieferung neuer Versionen bereits existierender Daten

Bei Datenlieferungen kann es vorkommen, dass neue Versionen bereits im Data-Store existierender Daten importiert werden müssen, um Datensätze zu korrigieren. In diesem Fall sind mehrere Möglichkeiten denkbar, die in der Konzeptphase noch weiter ausgearbeitet werden müssen.

Im einfachsten Fall werden die existierenden Daten überschrieben/gelöscht und durch die neuen Werte ersetzt. Dadurch existiert zu jedem Zeitpunkt nur eine Version des Datensatzes im Data-Store. Die ETL-Prozesse könnten für diesen Fall so vorbereitet sein, dass existierende Einträge vor dem Schreiben neuer Daten gelöscht werden. Dieser Ansatz setzt voraus, dass jede Lieferung einen vollständigen Datensatz beinhaltet. Da zu jedem Datensatz Data-Outputs definiert sein könnten, ist dieser Ansatz nur mit einer guten manuellen Qualitätssicherung nutzbar, bei der sichergestellt werden muss, dass nicht zu viele Daten gelöscht werden, die nicht durch die neue Lieferung wieder befüllt werden. Das Löschen von Datensätzen, würde in diesem Fall dazu führen, dass der Data-Output nicht mehr nutzbar wäre.

Sollen mehrere Versionen vorgehalten werden, müsste der Import zusätzlich zu den bereits existierenden Daten ausgeführt werden, um eine neue Version des Datensatzes zu erzeugen. Hierzu müssen die Datensätze mit einem Versionsstand (zum Beispiel durch ein Datum und/oder eine Versionsnummer) versehen werden. Beim Auslesen der Daten aus dem Data-Store muss dafür gesorgt werden, dass pro Datensatz nur Daten aus einem bestimmten Versionsstand verwendet werden. Über entsprechende Sichten (Views) in der Datenbank könnten alle Versionen aufgelistet und zur Auswahl bereitgestellt werden. Gleichzeitig könnten auch feste Sichten auf bestimmte Versionsstände definiert werden. Dieser Ansatz ist vor allem für konsistente Data-Outputs gut geeignet. Durch die Verwendung von stabilen Versionsständen würden sich Data-Output nicht automatisch durch neue Datenlieferungen ändern. Durch dieses Vorgehen könnten beliebig viele Versionsstände abgelegt werden. Während des Import-Prozesses müsste der Versionsstand mit angegeben werden.

3.2.3 Datenübermittlung

Für die Datenübermittlung zur Redaktion sind mehrere Möglichkeiten denkbar.

- Manuelle Übermittlung (zum Beispiel via E-Mail) von Dateien an die Redaktion:
  - Diese Möglichkeit gestaltet sich analog zu dem Austausch mit der bisherigen Excel-Tabelle. Daten werden dateibasiert an die Redaktion übergeben (zum Beispiel via E-Mail). Der ETL-Prozess muss nun manuell durch Mitarbeitende der Redaktion gestartet werden.
- Übermittlung der Dateien über eine Web-Formular
  - Anstatt der manuellen Übermittlung der Dateien an die Redaktion werden die Dateien über ein Web-Formular durch die entsprechenden datenhaltenden Stellen hochgeladen.

Datensatz-Parameter (zum Beispiel Metadaten) könnten über ein Formularfeld ergänzt werden. Anschließend wird direkt der ETL-Prozess mit dem gelieferten Datensatz ausgeführt, wodurch keine Arbeit durch die Redaktion notwendig ist.

- Automatisierung der Import-Prozesse:
  - Daten mit einem hohen Aktualisierungszyklus und einer festen Datenstruktur könnten automatisiert importiert werden. Hierzu muss die aktuelle Version der Datenquelle zunächst automatisch abgefragt werden können (zum Beispiel aus einer Datenbank, dateibasiert oder über einen Web-Service). Ein entsprechender Ansatz zur Versionierung ist in Kapitel 3.2.2 beschrieben. Ein automatischer Import setzt weiterhin voraus, dass die entsprechenden Metadaten entweder maschinell ausgelesen werden können oder dass diese nur beim ersten Erzeugen des Prozesses gepflegt werden müssen. In ETL-Server-Prozessen werden üblicherweise die Verwendung von zeitlichen Ablaufplänen angeboten. Das bedeutet, dass zuvor definierte ETL-Prozesse in bestimmten zeitlichen Intervallen oder zu vordefinierten Zeitpunkten automatisch gestartet werden.
  - Sollten sich die jeweiligen Datenstrukturen häufig ändern, ist von einer automatisierten Ausführung abzuraten, da entsprechende ETL Prozesse bei jeder Änderung der Struktur entsprechend angepasst werden müssen.

3.2.4 Softwareauswahl

Für die in Kapitel 3.2 diskutierten Anforderungen können verschiedene ETL Produkte verwendet werden. Im Folgenden werden zwei Varianten erläutert.

3.2.4.1 FME

Die Software FME (Feature Manipulation Engine) bestehend aus FME Desktop und FME Server des Herstellers Safe Software ist im UBA bereits lizenziert und in Verwendung. Zudem werden durch FME alle Anforderungen (s. Kapitel 2.5) zur Integration der Daten in den Data-Store abgedeckt. FME hat eine integrierte Unterstützung für zahlreiche Formate und Anwendungen sowie Transformationswerkzeuge, die es dem Nutzenden ermöglichen, benutzerdefinierte Integrations-Workflows zu erstellen und zu automatisieren. Die Besonderheit dabei ist die graphische Benutzeroberfläche, so dass komplexe Abläufe ohne besondere Kenntnisse von Programmiersprachen abbildbar sind. Weiterhin ist die Technologie auf eine Server Installation erweiterbar. FME Server bietet weiterführende Optionen der Automatisierung und damit letztlich Reduzierung der manuellen Arbeit, die bislang nötig ist.

3.2.4.2 Tableau

Im Rahmen des vorliegenden Berichtes ist die BI-Software Tableau eines der Tools, die hinsichtlich der komplexen Anforderungen analysiert werden und potenziell im Kontext des DataCube im UBA implementiert werden könnten. Während die Lösung basierend auf Tableau in 3.5.5 genauer betrachtet wird, wird an dieser Stelle auf die Komponente Tableau Prep hingewiesen, die Datenvorverarbeitungsschritte abbildet. Mit der Teilkomponente Tableau Prep Builder werden Datentransformationen, Bereinigungsschritte sowie das Zusammenführen von Daten realisiert (Tableau Software, LLC, o. J. - k). Die zweite Teilkomponente Tableau Prep Conductor setzt auf diese Funktionalitäten auf und bietet Methoden zur Automatisierung und Überwachung dieser Aufbereitungsprozesse (Tableau Software, LLC, o. J. - k).

3.2.4.3 Toolauswahl

Die Auswahl des konkreten ETL-Tools kann erst nach der Auswahl der Lösungskomponenten (s. Kapitel 3.5) erfolgen.

Tableau Prep wird als eine Alternative zur ETL-Software FME gesehen. Falls Tableau für den Einsatz im Data Cube Kontext ausgewählt wird, kann es sinnvoll sein, nicht nur für die Datenexploration auf die Tableau Produktpalette aufzusetzen, sondern ebenso die Softwarekomponenten von Tableau Prep für die Datenaufbereitung einzusetzen. Hierdurch wird einerseits die Kommunikation verschiedener Komponenten des DataCube untereinander gefördert sowie eine höhere Benutzerfreundlichkeit gesehen, da die einzelnen DataCube Konzepte auf Basis der gleichen Produktreihe implementiert werden könnten.

Für alle anderen Lösungskomponenten kann FME sowohl auf Grund der Abdeckung der Anforderungen als auch durch die existierende Produkterfahrung als ETL Tool empfohlen werden.

3.3 Data-Output und Data-Explorer

Basierend auf den fachlichen Anforderungen (s. Kapitel 2) werden existierende Software-Produkte evaluiert, um die Komponenten Data-Explorer und Data-Output umzusetzen.

Zur Evaluierung müssen die Funktionalitäten der existierenden Software-Produkte gegen alle bekannten Anforderungen abgeglichen werden. Hierzu werden Produktbeschreibungen, Dokumentationen sowie ergänzende Internetrecherchen der jeweiligen Produkte verwendet, um die Liste der Anforderungen abzugleichen. Die Anzahl der zu evaluierenden Produkte ist auf 4-5 beschränkt und wird vorher mit dem UBA abgestimmt.

Da der Workshop zum Thema technische Rahmenbedingungen durch das UBA (s. Kapitel 2.5) noch nicht stattgefunden hat, sind die technischen Anforderungen noch nicht final analysiert. Diese werden nach dem entsprechenden Termin in der Anforderungs-Matrix ergänzt.

Die folgenden Kriterien sollen bei der Evaluierung berücksichtigt werden:

- Abdeckung der Anforderungen (fachlich und technisch) aus den Workshops
- Anpassbarkeit
- Aufwand für Implementierung, Realisierung und Inbetriebnahme
- Dokumentation
- Marktreife und Realisierbarkeit in AP6
- Open-source Community
- Integration in die UBA-Webseite / Technische Kompatibilität
- Kosten / Wirtschaftlichkeit

Nach der Recherche über bestehende Software-Komponenten muss diese gemeinsam mit dem UBA abgestimmt werden. Sollte eine geeignete Software gefunden werden, kann die weitere Konzeption mit dieser durchgeführt werden. Falls keine geeignete existierende Software gefunden werden sollte, kann über eine eigene Implementierung nachgedacht werden. In diesem Falle ist eine Priorisierung der Anforderungen notwendig, um konkrete Entwicklungen zu konzipieren, die in dem gegebenen zeitlichen Rahmen umgesetzt werden können.

3.4 Redaktioneller Prozess

Während des Projektverlaufs wurde entschieden, dass der Fokus des redaktionellen Prozesses (s. Kapitel 2.1) im Data Cube Projekt auf dem Austausch und der Veröffentlichung von Daten auf der Webseite liegen soll. Die Redaktion von textuellen Inhalten ist daher nicht Teil der Konzeption. Durch das Data Cube Projekt werden sich einzelne Arbeitsschritte wie der Umgang mit Daten und der Erzeugung von Abbildungen für die Webseite ändern. Hierzu ist es notwendig, den Prozess mit den neuen Komponenten entsprechend anzupassen.

3.4.1 Integration von Data-Input, Data-Explorer und Data-Output in den bestehenden Prozess

Die minimalen Anpassungen, die für den redaktionellen Prozess vorgeschlagen werden, sind wie folgt:

Daten werden, soweit möglich, nicht mehr über die Excel-Templates ausgetauscht, sondern wie in Kapitel 3.2 beschrieben über den Data-Input in den Data-Store eingespielt. Da die Software für Data-Explorer und Data-Output noch nicht final definiert ist, kann hier noch keine exakte Beschreibung des Prozesses beschrieben werden. Es ist jedoch denkbar, dass die im Data-Explorer definierten Data-Outputs bereits vor der Einbettung im CMS mit allen Autoren zur Ansicht geteilt (zum Beispiel als URL) und somit unabhängig vom Text erstellt und bearbeitet werden können.

In den Workshops wurde definiert, dass neue Daten erst nach einer Freigabe durch die Redaktion veröffentlicht werden dürfen. Es gilt hier noch zu definieren, ob die Freigabe des Data-Outputs ausreichend ist, oder ob die Freigabe für die Datenlieferung im Data-Store erfolgen muss. Dies ist unter anderem abhängig davon, ob jede Datenlieferung als eigenständige Version definiert ist oder ob Tabellen fortlaufend weitergeführt werden (s. Kapitel 3.2.2). Bei einer fortlaufenden Weiterführung von Tabellen würden sich bereits existierende Data-Outputs automatisch anpassen, wenn neue Daten eingespielt werden. Über einzelne Versionsstände könnten Data-Outputs für einen bestimmten Versionsstand „eingefroren“ werden.

Sind Texte und Data-Outputs abgestimmt, kann ein Eintrag im CMS vorgenommen werden. Wie genau die Einbindung der Data-Outputs aussieht, kann ohne die konkrete Definition der Software ebenfalls noch nicht definiert werden. Möglichkeiten wären zum Beispiel die Einbindung über iFrames als HTML oder der direkten Verwendung von Drupal-Erweiterungen. iFrames erlauben eine CMS-neutrale Implementierung, wobei evaluiert werden muss, ob die entsprechenden Webseitenelemente responsiv eingebettet werden können. Durch Drupal-Erweiterungen könnten Data-Outputs nativ eingebunden werden. Die Auswahl möglicher Softwareprodukte wird hierbei jedoch stärker eingeschränkt, bzw. wird der Aufwand zur Implementierung deutlich erhöht. Auch die Verlinkung von externen Webseiten, zum Beispiel für explorative Tools mit vielen Konfigurationsmöglichkeiten muss evaluiert werden. Dieser Ansatz bietet die größte Flexibilität zur Integration von verschiedenen Produkten. Gleichzeitig bedeutet eine reine Verlinkung jedoch auch einen Medienbruch für den Anwendenden, da das jeweilige Tool erst nach einem Wechsel der Webseite verwendet werden kann.

Für die Erstellung der statischen Graphiken kann voraussichtlich eine Export-Funktion des Data-Outputs oder Data-Explorers verwendet werden.

3.4.2 Weitere Möglichkeiten zur Optimierung

Aktuell ist die Organisation der Daten und zugehörigen Artikeln und Berichten durch eine gute Organisation einer zentralen Excel Tabelle möglich. Innerhalb dieser werden sowohl Inhalte

einzelner Artikel und Berichte als auch Zeitpunkte sowie Verantwortlichkeiten festgehalten. An dieser Stelle sind mehrere Ansätze denkbar, um einzelne Prozessschritte zu optimieren:

- Durch die Verwendung klarer Data-Input Prozesse können automatisierte Benachrichtigungen an die Redaktion versendet werden, sobald neue Daten vorliegen. Diese Information könnte unter anderem auch automatisch in der Steuerungstabelle eingetragen werden.
- Hierfür könnte die Excel Tabelle weiterhin verwendet werden. Häufig auftretende Nachteile bei Excel Tabellen für automatisierte Prozesse sind unter anderem: Konflikte bei parallelen Zugriffen, keine/wenig Validierung der Dateneingaben.
- Alternativ hierzu könnte man überlegen, die Steuerungsdaten auch in eine Datenbank zu überführen. Dies würde eine Implementierung einer Editor-Funktion (ggf. auch im Data-Explorer) voraussetzen. Vorteile wären jedoch, dass weitere Automatisierungsschritte möglich wären: automatische Validierungen der Eingaben, automatische Historisierung der Einträge, interaktive Verlinkung zu den konkreten Datensätzen im Data-Store.

3.5 Evaluation möglicher Lösungskomponenten

Als Grundlage für die Auswahl einer Softwarelösung für das Data Cube Projekt werden in diesem Kapitel fünf mögliche Technologien beschrieben und die damit verbundenen Lösungswege skizziert. Bevor in den nächsten Unterkapiteln die tool-spezifischen und detaillierten Anforderungsbewertungen folgen, wird zunächst das generelle Vorgehen erläutert.

Auf Basis der Ergebnisse des Kapitels 2 wurde ein Anforderungskatalog erstellt (vgl. Anhang A.1). In diesem sind die gesammelten Spezifikationen an die verschiedenen Data Cube Komponenten aufgeführt. Zudem wurde ein einheitliches Bewertungsschema entwickelt, um eine gewisse Vergleichbarkeit der später zu analysierenden Lösungen zu ermöglichen. Hierbei wird zunächst eine generelle Einordnung des Abdeckungsgrades der jeweiligen Anforderung ermittelt. Anschließend folgt eine kurze Beschreibung der Umsetzung im jeweiligen Tool, die dann wiederum durch Angabe einer möglichen Referenzquelle ergänzt wird. Für die Bewertung des Abdeckungsgrades wurden die Kategorien

- Anforderung ist erfüllt (+),
- Anforderung ist nicht erfüllt, doch eine Erweiterung (z. B. durch Eigenentwicklung) ist prinzipiell denkbar (o),
- Anforderung kann nicht, oder nur mit sehr hohem Aufwand, abgedeckt werden (-)

ausgewählt. Durch diese drei Kategorien, die durch die in Klammern stehenden Zeichen symbolisiert werden, soll eine Transparenz über den Abdeckungsgrad einer jeden Anforderung für die verschiedenen Lösungsansätze gewährleistet werden. Zudem wird in diesem Kontext die Auswahl einer lediglich dreistufigen Bewertungsskala als zielführend erachtet, um die durchaus gegebene Komplexität der vollständigen Anforderungstabelle nicht unnötig zu erhöhen und dadurch eine objektive Bewertung zu erschweren. Ergänzend zu den drei Kategorien wird darauf hingewiesen, dass, wenn eine Anforderung generell nicht in dem Kontext der jeweiligen Lösung gesehen wird, der Vermerk „Unabhängig von [Name der Lösungskomponente]“ verwendet wird und in diesem Fall die Spalte für den Abdeckungsgrad nicht ausgefüllt wird.

Da aus Platzgründen einige Spalten der Tabelle ausgelassen wurden, können die einzelnen Komponenten in den Anforderungstabellen nur noch über die Spalte „Nummer“ identifiziert werden. Die Abkürzungen stehen dabei für: Allgemeine Anforderungen (AA), Data-Output (DO),

Data-Explorer (DE), Data-Store (DS) und Data-Input (DI). Die Beachtung ist vor allem wichtig, da sich einige Funktionalitäten zwischen Data-Output und Data-Explorer überschneiden können. Wichtig ist, dass nur die Anforderungen des Data-Outputs für externe Nutzende gelten.

Bevor der detaillierte Vergleich mittels der vollständigen Bewertungstabelle für fünf mögliche Lösungskonzepte erfolgte, lag der Fokus auf einer Vorauswahl dieser fünf Lösungen. Für diesen Prozess wurde eine reduzierte Liste der Anforderungskriterien verwendet, die für insgesamt 15 Lösungen ausgefüllt wurde (vgl. Anhang A.3). Ebenso wurden von diesen bereits die Tools Sisense, .Stat Suite, Highcharts, Tableau, Metabase und Socrata / Tylersoft in einem Dokument hinsichtlich besonderer Stärken und Schwächen verglichen (vgl. Anhang A.4).

Aufgrund der zahlreichen Lösungskomponenten, die im Kontext der Internetrecherche für den Data Cube gefunden wurden, kristallisierte sich dieses mehrstufige Verfahren als sinnvoll und zielführend heraus. Auf Basis der Ergebnisse für die Tool-Vorauswahl (s. Anhang A.3 und Anhang A.4) konnte somit eine fundierte Entscheidung für die Auswahl der Lösungen .Stat Suite, Highcharts, Mesap, Sisense und Tableau erfolgen. In den nachstehenden Kapiteln sind diese Lösungsansätze separat thematisiert. Der ausgefüllte Anforderungskatalog ist jeweils in tabellarischer Form beigefügt. Auf Basis der tool-spezifischen Bewertungen folgt in Kapitel 3.5.6 ein abschließender Vergleich der Lösungsvorschläge.

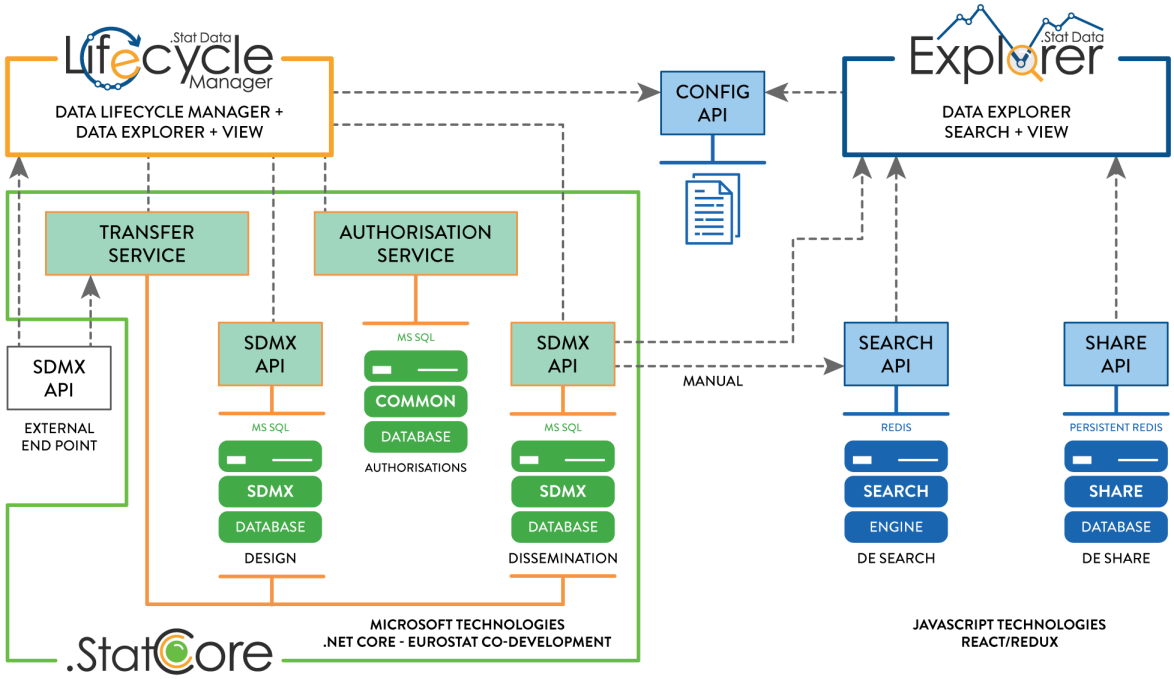
Die Darstellung der einzelnen Möglichkeiten sowie der Vergleich kann als Entscheidungsgrundlage für das UBA verwendet werden. Im weiteren Verlauf der Konzeption soll eine Lösungskomponente oder eine Kombination aus mehreren Komponenten für die Umsetzung des Data Cube Projektes durch das UBA ausgewählt werden. Anschließend soll ein detaillierterer Lösungsansatz beschrieben werden, der in der Implementierungsphase umgesetzt werden soll. Der Lösungsansatz wird nach der Auswahl in Kapitel 4 beschrieben.

3.5.1 .Stat Suite

3.5.1.1 Lösungsansatz

Die .Stat Suite ist eine open-source-Plattform mit einem Schwerpunkt auf der Bereitstellung und Visualisierung statistischer Daten. Sie ist kein monolithisches Programm, sondern besteht aus drei, über APIs lose gekoppelten, verbundenen Komponenten. Im Folgenden wird zunächst die Funktion und das Zusammenspiel dieser Komponenten beschrieben. Dann wird erläutert, wie sie zur Umsetzung des Data Cube Konzepts eingesetzt werden können.

Abbildung 5: Die Architektur der .Stat Suite



Quelle: (Statistical Information System Collaboration Community - SIS-CC, o. J. - i)

Die .Stat Suite besteht aus den Komponenten .Stat Data Lifecycle Manager (Abkürzung: .Stat DLM), .Stat Data Explorer (Abkürzung: .Stat DE) und .Stat Core. Im .Stat Core werden Daten und Metadaten gespeichert, verwaltet und über APIs bereitgestellt. Der .Stat DLM setzt auf den .Stat Core auf. Er bietet authentifizierten Benutzenden eine Weboberfläche für den Upload und die Verwaltung von Daten und Metadaten. Der .Stat DE ist die Weboberfläche für den öffentlichen Zugriff auf die Daten. Er bietet Funktionen für das Suchen und Teilen von Datensätzen, zeigt Datensätze als Tabellen und in verschiedenen graphischen Visualisierungen an.

Diese drei Komponenten sind über standardisierte APIs lose gekoppelt. Wie bei einem Baukasten kann ein Einsatz der .Stat Suite theoretisch beliebige dieser Komponenten miteinander kombinieren. Ebenso können einzelne Komponenten durch andere Produkte ersetzt werden, solange diese die entsprechenden APIs unterstützen. Für den Lösungsansatz, basierend auf der .Stat Suite, wird eine vollständige Installation der Gesamtplattform inklusive aller drei Komponenten vorgeschlagen. Eine nennenswerte Alternative wäre es, nur den .Stat DE zu installieren, ihn mit einer anderen Datenhaltung zu koppeln, welche die notwenigen APIs bereitstellt, und den .Stat DE als reines Veröffentlichungswerkzeug zu nutzen.

Die Koppelung der Komponenten erfolgt hauptsächlich über SDMX-APIs. Das Austauschformat SDMX umfasst eine Reihe von offenen Standards für die Beschreibung und den Austausch von statistischen Daten (Statistical Data and Metadata eXchange - SDMX Community, 2021 - b). Neben dem erwähnten API-Standard, der die Kommunikation zwischen den Komponenten ermöglicht, ist SDMX auch ein Dokumentformat, um Daten und Metadaten zu formatieren. Die .Stat Suite wird als “SDMX-native” beschrieben (Statistical Information System Collaboration Community - SIS-CC, o. J. - c).

Bereits der Data-Input ist abhängig von SDMX: Der Upload von Daten über den .Stat DLM ist nur in den Formaten SDMX oder als Excel Datei mit einer sogenannten „Excel data definition (EDD)“ XML-Datei möglich (Statistical Information System Collaboration Community - SIS-CC, o. J. - ab).

Zudem können nur solche Daten hochgeladen werden, für die bereits vollständige SDMX-Metadaten, sogenannte Strukturdefinitionen, vorliegen.

Um die Anforderungen an die Data-Input Komponente abzubilden, ist demnach ein ETL-Präprozess nötig, um:

- die Daten selbst in SDMX zu transformieren;
- die Daten in einer SDMX-Strukturdefinition zu beschreiben, die ihre Maßeinheiten, Dimensionen, etc. definiert;
- diese Strukturdefinition mit den bereits bestehenden abzugleichen, um eine homogene Datenhaltung aufzubauen.

Insbesondere Letztgenanntes ist eine umfassende Aufgabe, die eng auf den Informationsaustausch mit dem .Stat Core abgestimmt werden sollte.

Wie der Datenimport ist auch die Datenhaltung im .Stat Core, deren Bedienung im .Stat DLM geschieht, auf das SDMX-Format konzentriert. Somit wäre beim Einsatz der .Stat Suite Plattform der SDMX-Standard notwendigerweise auch für die Data-Store Komponente zu berücksichtigen und zu implementieren. SDMX bietet eine umfassende Struktur, um Dimensionen und Werte präzise zu definieren. Komplexe Schachtelungen von Dimensionen innerhalb eines Datensatzes sind möglich. Ebenso wird die Verwendung derselben Dimension in beliebig vielen Datensätzen unterstützt (Statistical Data and Metadata eXchange - SDMX Community, 2021 - e). Im .Stat DLM lassen sich die im .Stat Core gespeicherten Dimensionen sowie die Abhängigkeiten und Hierarchien zwischen ihnen betrachten (Statistical Information System Collaboration Community - SIS-CC, o. J. - w).

Die Datenhaltung kann in mehreren Data Spaces erfolgen. Data Spaces sind logische Unterteilungen bzw. Gruppierungen von Daten, die wie separate Datenbanken verstanden werden können und für die jeweils unterschiedliche Freigaben möglich sind. In Abbildung 5 sind zum Beispiel zwei voneinander unabhängige Bereiche für Design und die Verteilung der Daten (engl.: „dissemination“) skizziert. Dieser Mechanismus kann genutzt werden, um einen Workflow für die Qualitätskontrolle aufzubauen. Innerhalb des .Stat DLM gibt es Werkzeuge, um Daten und Strukturen zwischen Data Spaces zu kopieren.

Über die persistente Speicherung der Datensätze und ihrer Metadaten sowie die Bereitstellung über APIs hinaus ist die Funktionalität der Datenhaltung in der .Stat Suite gering: Eine Historisierung der Daten wird nicht unterstützt. Eine Versionierung kann durch das Format SDMX ausgedrückt werden, die .Stat Suite bietet allerdings keine Werkzeuge, um die dazu in den importierten Daten definierten Angaben zu bearbeiten. Auch das Bearbeiten der Datensätze und ihrer Dimensionen ist nicht möglich.

Da alle Komponenten open-source sind, besteht grundsätzlich die Möglichkeit, durch eigene Zusatzleistungen den .Stat Core sowie den .Stat DLM um weitere Funktionalitäten bzw. Bedienoberflächen zu erweitern, um die bisher offen gebliebenen Anforderungen an die Data-Store Komponente abzubilden. Eine Anpassung am Quellcode führt in der Regel jedoch zu mehr Arbeit, falls ein Update der Software installiert werden soll.

Der .Stat DE ist die Komponente für den Data-Output. Er bietet eine eigenständige Weboberfläche, die Nutzenden offensteht. Datensätze können mithilfe einer durchdachten graphischen Benutzeroberfläche mit vielen Steuerungsmöglichkeiten gesucht werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - r). Zudem werden zahlreiche Konfigurationsmöglichkeiten der Oberfläche bereitgestellt, wodurch zum Beispiel

bestimmte Design Vorgaben abgedeckt werden könnten (Statistical Information System Collaboration Community - SIS-CC, o. J. - g). Für die Datenrecherche steht eine facettierte Suche bereit (Statistical Information System Collaboration Community - SIS-CC, o. J. - o). Weiterhin wird eine Reihe von Visualisierungsmöglichkeiten zur Verfügung gestellt. Tabellen können gefiltert und sortiert werden. Durch die sogenannte „sharing"-Funktionalität ist es möglich, fest definierte Ansichten zu Tabellen und Visualisierungen zu teilen. Hervorzuheben ist ein erkennbares Bemühen um eine barrierearme Bedienung (Statistical Information System Collaboration Community - SIS-CC, o. J. - n).

Für die Darstellung von interaktiven Diagrammen ist für externe Nutzende die .Stat DE Komponente vorgesehen. Eine native Integration in Drupal als Data-Outputs ist jedoch nicht vorhanden. Zur Integration von interaktiven Diagrammen zum Beispiel in Drupal-Artikeln müsste daher ein eigenes Drupal-Modul implementiert werden. Hierzu könnten gegebenenfalls Komponenten aus dem Quellcode des .Stat DE übernommen werden. Alternativ kann die SDMX-API verwendet werden, um eigenständige Data-Output Komponenten zu entwerfen. Eine Definition von Dashboards ist in der .Stat Suite ebenfalls nicht vorgesehen. Auch hierzu müssten eigene Komponenten implementiert werden. Je nach Implementierung der Data-Outputs könnte die jeweilige Methodik auch für die Kombination verschiedener Diagramme zu Dashboards verwendet werden.

3.5.1.2 Anforderungstabelle

In der folgenden Tabelle werden alle Anforderungen an den Data Cube durch die Lösungskomponente beschrieben und nach der Klassifikation aus Kapitel 3.5 bewertet.

Tabelle 2: Anforderungstabelle der Lösungskomponente .Stat Suite

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
AA1	Die Software soll erweiterbar/ anpassbar sein, falls nicht alle Anforderungen abgedeckt werden können (Anpassung von Design, aber auch Funktionalitäten)	+	Als Open Source Projekt insgesamt flexibel anpassbar. Ein zusätzlicher Vorteil ist die Verbindung des Datenflusses der drei Hauptkomponenten (Core, DLM, DE) durch standardisierte SDMX-APIs. Dadurch kann eine eigene Architektur geschaffen werden, die aus den drei Hauptkomponenten beliebige nutzt und beliebige durch andere Komponenten ersetzt. Solange SDMX-APIs bereitgestellt werden, sind verschiedene Architekturen möglich.	https://sis-cc.gitlab.io/dotstatsuite-documentation/getting-started/framework/#introduction
AA2	Die Anforderungen müssen durch die Software effizient implementiert werden können.	-	Die Implementierung ist komplex aufgrund der vielen Komponenten, die auf verschiedenen Technologien beruhen (z.B. sind zwei Datenbanktechnologien im Einsatz: Postgres und	siehe z.B. https://sis-cc.gitlab.io/dotstatsuite-documentation/getting-started/infrastructure-requirements/

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			Microsoft SQL-Server). Spätere Aktualisierungen einzelner Komponenten werden dementsprechend ebenfalls komplex.	
AA3	Die Software soll mit geringem Aufwand in Betrieb genommen werden.	-	Aufgrund der Komplexität des Systems (vgl. AA2) ist ein hoher Aufwand für die Inbetriebnahme zu erwarten.	
AA4	Die Software muss für Anwender und Administratoren ausreichend dokumentiert sein.	o	Eine Dokumentation ist in einer Reihe von Artikeln auf der Webseite vorhanden, allerdings in begrenzter Detailtiefe. Für ausgewählte Themen stehen Webinar Mitschnitte zur Verfügung.	Dokumentation auf der Webseite siehe: https://sis-cc.gitlab.io/dotstatsuite-documentation/ Webinar Mitschnitte siehe: https://www.youtube.com/channel/UCZGlYlrmeb1MbLONpxObGUQ/videos
AA5	Die Software ist bereits ausgereift und kann direkt verwendet werden.	+	Die .Stat Suite wird für verschiedene (oft internationale) Datenportale eingesetzt.	https://sis-cc.gitlab.io/dotstatsuite-documentation/about/powered-by/
AA6	Es gibt eine aktive Community mit Diskussionsforen und Beispielen zur Anwendung.	o	Es existiert eine Community, die aber wenig sichtbar ist. Es gibt keine öffentlichen Foren oder Ähnliches.	Im weitverbreiteten Progammierforum „Stack Overflow“ finden sich kaum Diskussionen unter dem Tag „statsuite“ (https://stackoverflow.com/search?q=.stat+suite), die Webinar Mitschnitte und Videos der Dachorganisation auf YouTube haben meist unter 100 Views (https://www.youtube.com/channel/UCZGlYlrmeb1MbLONpxObGUQ/videos).
AA7	Die Software kostenfrei nutzbar, oder die Kosten sind in einem für das	+	Der Software-Code ist open-source und steht unter der MIT Lizenz kostenlos zur Verfügung. Der Betrieb beruht jedoch an einigen Stellen auf	siehe z.B. https://sis-cc.gitlab.io/dotstatsuite-documentation/getting-started/infrastructure-requirements/

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Projekt vertretbaren Rahmen.		Microsoft-Produkten (SQL-Server und IES). Diese sind in kostenfreien Versionen verfügbar, für produktive Nutzungen müssen hier jedoch verschieden Lizenzen evaluiert werden.	
AA8	Die Software kann im UBA Rechenzentrum betrieben werden	+	Eine lokale Installation ist auf Basis des Sourcecodes selbst oder auf Basis von Containertechnologie (Docker) möglich.	https://sis-cc.gitlab.io/dotstatsuite-documentation/install-source-code/ https://sis-cc.gitlab.io/dotstatsuite-documentation/install-docker/ https://sis-cc.gitlab.io/dotstatsuite-documentation/install-service/
AA9	Die Software kann ohne Weiteres gratis getestet werden	+	Grundsätzlich ist die gesamte Nutzung kostenlos (siehe AA7). Zusätzlich ist eine vorbereitete Demoversion mit Webinar-Video kostenfrei verfügbar.	https://www.youtube.com/watch?v=U2knnqOr5ws
DO1	Es muss eine Visualisierung von Daten möglich sein, bei denen eine feste Ansicht auf die Daten und eine feste Darstellungsform durch die Redaktion vorgegeben wird.	o	Durch die „sharing“-Funktionalität des DE ist es möglich, fest definierte Ansichten, Tabellen und ihre Visualisierungen als sowohl „embedded Code“ (iFrame) als auch durch eine URL zu teilen. Realisiert in Komponente DE. Für den DO muss voraussichtlich ein eigenes Drupal Modul auf Basis der SMDX-API implementiert werden.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/share/
DO2	Daten müssen als Tabelle dargestellt werden können.	o	Tabellen sind die Standardansicht im DE. Für den DO muss voraussichtlich ein eigenes Drupal Modul auf Basis der SMDX-API implementiert werden.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/
DO3	Daten innerhalb der Tabellendarstellung sollen gefiltert werden können.	o	In Komponente DE möglich. Für den DO muss voraussichtlich ein eigenes Drupal Modul auf Basis der SMDX-API implementiert werden.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DO4	Daten innerhalb der Tabellendarstellung sollen sortiert werden können.	-	Im DE können nur die Tabellenspalten sortiert werden, nicht aber die Inhalte.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/filters/
DO5	Aggregationen von Datensätzen sollen vorgenommen oder vorberechnete Aggregationen angezeigt werden können.	-	Anzeige ist im DE möglich. Aggregationen müssen außerhalb des Systems berechnet, in SDMX formatiert und beschreiben sowie eingespielt werden (siehe Anforderungen DS für Details). Für den DO muss voraussichtlich ein eigenes Drupal Modul auf Basis der SMDX-API implementiert werden.	
DO6	Deaggregationen von Datensätzen sollen vorgenommen oder vorberechnete Deaggregationen angezeigt werden können.	-	Anzeige ist im DE möglich. Deaggregationen müssen außerhalb des Systems berechnet, in SDMX formatiert, beschrieben sowie eingespielt werden (siehe Anforderungen DS für Details).	
DO7	Zeitreihen sollen eingeschränkt werden können (Filterung durch min. und max. Datum).	o	Zeit kann wie alle definierten Dimensionen als Filter verwendet werden. Realisiert in Komponente DE. Für den DO muss voraussichtlich ein eigenes Drupal Modul auf Basis der SMDX-API implementiert werden.	
DO8	Einheiten sollen dynamisch umgerechnet werden, oder durch vordefinierte Daten in anderen Einheiten angezeigt werden können.	o	Für vordefinierte Daten durch Flexibilität im Umgang mit Dimensionen umsetzbar. Die Darstellung erfolgt anschließend wie in DO1. Eine dynamische Umrechnung im Data-Output ist nicht möglich.	
DO9	Die Darstellungsform der Abbildungen soll durch den Nutzenden frei gewählt werden	o	Folgende Darstellungen sind möglich und durch die Nutzenden frei wählbar: Timeline, Choropleth Map, Symbol Chart (vertikal und horizontal), Scatterplot, Bar	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/charts/

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	können. Gewünschte Darstellungsformen sind: Liniendiagramm, Balkendiagramm, Tortendiagramm, Baumdiagramme, Abweichungen, Korrelationen und Streudiagramme, Häufigkeitsverteilungen, Nominale Vergleiche (wie Blasendiagramme oder Heatmaps), Fluss und Sankey-Diagramme, Netzwerkdiagramme)		Chart, Row Chart und Stacked Bar Chart. Realisiert in Komponente DE. Für den DO muss voraussichtlich ein eigenes Drupal Modul auf Basis der SMDX-API implementiert werden.	
DO10	Die Visualisierungen in Diagrammen/Abbildungen (Data-Outputs) müssen direkt in Artikel auf der UBA-Webseite (Drupal CMS) eingebunden werden können	-	Im Standard ist die Integration nur als iFrame („embedded code“) oder als URL aus dem share-service (siehe DO1) möglich. Eine native Integration im CMS müsste implementiert werden.	
DO11	Die Visualisierungen in Diagrammen/Abbildungen (Data-Outputs) soll in der zukünftigen umwelt.info Webseite möglich sein.	-	siehe DO10	
DO12	Die Implementierung soll möglichst CMS-offen (unabhängig von dem bestehenden Drupal System) erfolgen. Es soll geprüft werden	o	Einbettung als „embedded code“ und als URL geschieht CMS-unabhängig über share-service (siehe DO1). Alternativ können auch die SDMX-APIs als Datenquellen für eine eigens zu programmierende (Web-) Oberfläche eines CMS	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	ob Inhalte z.B. als responsive iFrames oder vergleichbares eingebettet werden können.		genutzt werden. Realisiert durch DE und Core.	
DO13	Funktionalitäten die nicht CMS-offen implementiert werden, sind als Drupal-Modul (unter Beachtung der Drupal Code-Konventionen) zu entwickeln in vollständig lauffähig in das CMS der UBA Webseite zu integrieren.	o	siehe DO12	
DO14	Es muss möglich sein, Dashboards im Drupal CMS einzubinden, die aus mehreren Data-Outputs bestehen	o	Im DE werden nur einzelne DO bereitgestellt. Eine Dashboard Funktion existiert im Standard nicht und müsste zusätzlich implementiert werden. Die Integration der DO ins CMS ist zusätzlich unklar. Es ist denkbar, eigene DO und Dashboards basierend auf den SDMX-APIs für Drupal zu implementieren.	
DO15	Dashboards und Data-Outputs sollen auf verschiedenen Display Größen (Mobilgeräte und größere Bildschirme) gut angezeigt werden.	o	Der DE ist responsiv. Die DO Komponente ist noch unklar.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/design-principles/#responsiveness
DO16	Daten und Abbildungen sollen in unterschiedlichen Formaten zum Download angeboten werden. Daten: CSV, Excel	o	Die DO Komponente ist noch unklar. Im DE Daten können über das Suchen-Interface als Excel oder SDMX-CSV heruntergeladen werden. Ein Download von Rasterdateien ist nicht möglich.	https://sis-cc.gitlab.io/dotstatsuite-documentation/configurations/de-configuration/#enabled-download-option-on-the-search-result-page

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Abbildungen: PNG/JPEG			
DO17	Datensätze die zu einem Thema gehören sollen erkundet werden können. Das heißt, dass die Daten sowie verlinkte Datensätze angezeigt werden.	o	Die DO Komponente ist noch unklar. Im DE gibt es ein gutes Suchen-Interface mit vielen Steuerungsmöglichkeiten auf Basis der Metadaten zu den Dimensionen (= SDMX-Strukturdefinitionen). Es existiert keine Anzeige von verwandten Datensätzen.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/searching-data
DO18	Verknüpfte Datensätze sollen in der Visualisierung mit hinzugefügt werden können, um Daten vergleichen zu können.	-	Im DE nicht möglich. Bei einer eigenen DO Drupal Implementierung könnte diese Funktion berücksichtigt werden. Es existiert keine Standard-Funktionalität zum Verknüpfen von verwandten Datensätzen.	
DO19	Das Corporate Design ist bei der Gestaltung aller Nutzeroberfläche n zu beachten.	o	Die DO Komponente ist noch unklar. Die Oberflächen des Data Lifecylcle Manager (DLM) und des DE haben ein vorgegebenes Design mit sehr beschränkten Einstellungen. Grundsätzlich könnte der Programm-Code für weitere Anpassungen bearbeitet werden. Dies erschwert jedoch mögliche Updates.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/general-layout/
DO20	Es ist zu berücksichtigen wie Data-Outputs auch in die Datensuche des UBA eingebunden werden können.	o	Wenn Data-Outputs als Drupal-Module implementiert werden, kann der Standard Drupal-Mechanismus verwendet werden. Hierzu müssen beim Erstellen der Inhalte in Drupal entsprechende Keywords vergeben werden.	
DE1	Über den Data-Explorer müssen Datensätze des Data Stores auffindbar gemacht werden.	+	DE und DLM bieten hierzu Suchmöglichkeiten.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DE2	Metadaten von Datensätzen sollen angezeigt werden können.	+	Die Metadaten (= die SDMX-Strukturdefinitionen) können im DLM betrachtet und in direkten Abhängigkeiten zueinander dargestellt werden.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/list-related-data-structures/
DE3	Datensätze sollen als interaktive Diagramme angezeigt werden können.	+	Ist im DE möglich.	
DE4	Datensätze sollen tabellarisch angezeigt werden können.	+	Es stehen verschiedene Möglichkeiten zur Verfügung: (1) Datensätze können im Preview-Modus des DLM angezeigt werden. (2) Wenn sie bereits veröffentlicht sind, kann die Tabellenansicht des DE genutzt werden (siehe DO1 ff.). (3) Alternativ kann das Excel-Addon genutzt werden.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/preview-data/ und https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm-excel-addin/
DE5	Eine Definition von zusammengehörigen Datensätzen für einen Cube soll möglich sein.	+	Die Dimensionen der Datensätze werden in den Metadaten (= SDMX-Strukturdefinitionen) definiert und in der .Stat Suite im Bereich „structure“ des jeweiligen data space verwaltet. Die Verbindung zwischen Datensätzen erfolgt durch Nutzen derselben Dimensionen. Realisiert durch DLM.	
DE6	Es soll möglich sein, alle Cubes aufzulisten.	+	Im DE sind alle Datensätze sichtbar und suchbar. Über die SDMX-API kann mithilfe eines zu programmierenden Clients ebenfalls die Liste aller Datensätze abgefragt werden. (Zusatzleistung nötig)	
DE7	Es soll möglich sein alle Datensätze aufzulisten, die in einem Cube verwendet werden.	o	Im DLM können über "Related Structures" verwandte Datensätze aufgelistet werden. Im DE ist diese Funktionalität nicht möglich.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/preview-data/
DE8	Es soll möglich sein für einen Datensatz	-	Dazu gibt es kein Werkzeug. Der DLM ist stark auf die Verwaltung der Metadaten (=	https://sis-cc.gitlab.io/dotstatsuite-

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	anzuzeigen, in welchem Cube er verwendet wird.		SDMX-Strukturdefinitionen) konzentriert. Über die SDMX-API können die benötigten Informationen mithilfe eines zu programmierenden Clients abgefragt werden. (Zusatzleistung nötig)	documentation/using-dlm/dlm_overview/
DE9	Für jeden Datensatz sollen fachlich begründete Dimensionen ausgewählt werden können.	-	Dazu bietet die .Stat Suite kein Werkzeug. Die Metadaten (= SDMX-Strukturdefinitionen) zu den bereits definierten im System verwalteten Dimensionen können aber ausgelesen und einem zu bauenden Werkzeug verwendet werden (Zusatzleistung nötig).	
DE10	Hierarchische Datensätze sollen miteinander verknüpft werden können.	o	Hierarchien können nur über "Hierarchical Codelists" in SDMX abgebildet werden und müssen daher während des Data-Input angelegt werden.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/edit-structure/data-structure-wizard-example/#hierarchical-codelists
DE11	Datenauszügen für einzelne Datensätze sollen definiert werden können.	+	Durch die „sharing“-Funktionalität des DE gegeben (siehe DO1).	
DE12	Eine Konfiguration von Datensätzen / Dimensionen für einzelne Data-Outputs soll ermöglicht werden. Es soll konfigurierbar sein, welche weiteren Datensätze zu einem Data-Output potenziell hinzugeladen werden dürfen.	o	Der DE stellt alle Dimensionen eines Datensatzes dar und ermöglicht den Endnutzenden ein flexibles Auswählen. Auf diese Weise können sie einen Data-Output mit den für sie interessanten Dimensionen erzeugen. Freigabemanagement ist auf Ebene der Datensätze nicht möglich (s. RP1).	
DE13	Konfiguration der initialen und konfigurierbaren Darstellungsform	o	Der DE hat ein vorgegebenes Design mit wenigen Einstellungsmöglichkeiten (s. DO19). Da das Projekt Open Source ist, kann das Design mit	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			etwas Programmieraufwand beliebig angepasst werden. (Zusatzleistung nötig)	
DE14	Konfiguration weiterer Optionen des Data-Outputs für die Nutzenden: - Daten hinzuladen - Einheiten umrechnen / Datensatz mit anderen Einheiten laden - Aggregationen berechnen / Datensätze mit anderen Aggregationen laden - Erlaubte Downloads	-	Es existieren keine Konfigurationsmöglichkeiten für Data-Output Funktionalitäten für die Nutzenden.	
DE15	Für jeden Data-Output sollen die Zusatz-Funktionalitäten (z.B. Aggregationen berechnen, Daten hinzuladen, …) bei Bedarf deaktiviert werden können.	-	siehe DE14	
DE16	Konfiguration von Dashboards (Kombination aus mehreren Data-Outputs)	-	Keine Dashboard-Oberfläche oder Dashboard-Funktionalität, die mehrere Data-Outputs parallel anzeigen kann, vorhanden (siehe DO14). Der DE konzentriert sich auf das Finden und Zugänglichmachen einzelner Data Outputs.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/design-principles/
DE17	Der Data-Explorer soll außerhalb des UBA CMS implementiert werden können und hat kein spezifisches CMS zur Voraussetzung.	+	DLM und Core sind eigenständige Komponenten, die über SDMX-APIs miteinander kommunizieren können (und auch mit beliebigen anderen Komponenten, die mit einer SDMX-API kommunizieren können) (siehe AA1).	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DE18	Für externe Analysetools soll der Zugriff auf die Daten über eine API ermöglicht werden.	+	Für jeden Data Space stehen SDMX-APIs zur Verfügung. Realisiert durch den Core.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-api/api-main-features/
DE19	Für externe Analysetools soll der Zugriff auf die Daten über einen Export ermöglicht werden.	+	Export über manuellen Download im DE (siehe DO16)	
DE20	Die Konfiguration der Data-Outputs und Data Cubes soll über ein Webinterface möglich sein.	+	Alle Komponenten sind als Web-Anwendungen implementiert.	
DE21	Es soll möglich sein Zusatzdateien für Berichte (Bilder in verschiedenen Formaten, Daten) herunterzuladen.	+	Export über manuellen Download im DE (siehe DO16)	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/dump-mode/
DE22	Änderungen an den Daten sollen angezeigt werden können (Überarbeitungs modus)	-	Nicht möglich.	
DI1	Es soll eine Nutzeroberfläche und ein Workflow gestaltet werden um weitere Datenquellen anzubinden.	o	Der DLM ist als Benutzeroberfläche vorhanden. Uploads sind jedoch auf das SDMX-Format beschränkt (s. DI4). Werkzeuge für den Aufbau eines Workflow nicht vorhanden, allerdings besteht durch Vorhandensein der SDMX-APIs prinzipiell die Möglichkeit, solche Werkzeuge anzubinden.	
DI2	Der Data-Input soll entweder durch die bestehenden Excel Templates der Redaktion oder falls möglich	o	Es existiert ein Excel-Addin für den DLM, das sich auch für den Input neuer Daten verwenden lässt. Allerdings kann auf diese Weise kein Input von Strukturdefinitionen geschehen. Das Addin erlaubt	Excel-Addin: https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm-excel-addin/ Input Prozess: https://sis-cc.gitlab.io/dotstatsuite-

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	durch eine direkte Anbindung der Datenhaltung der datenhaltenden Stellen durchgeführt werden können.		nur Aktualisierung und Input von Daten in vorhandene Strukturen. Neue Strukturdefinitionen müssen per Upload im DLM importiert werden (s. DS11), bevor ein Input von Daten in diesen Strukturen möglich ist. Neben diesem Addin ist Input nur in SDMX-Format möglich, direktes Importieren aus Datenquellen nicht vorgesehen.	documentation/using-dlm/upload-data/
DI3	Daten sollen beim Import eine Qualitätssicherung durchlaufen	o	Eine Validierung findet beim Upload statt. Angesichts der Notwendigkeit von Präprozessen (siehe Anforderungen in DS), die den Upload in den DLM erst ermöglichen, ist diese Validierung allein nicht ausreichend.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/upload-data/data-validation/
DI4	Der Data-Input muss mit verschiedenen Eingangsformaten (Datenbanken, Excel, CSV, WebServices) umgehen können um Daten von verschiedenen datenhaltenden Stellen einzulesen.	-	Es wird nur das SDMX-Format (sowie xslx+edd) unterstützt. Datenquellen müssten also durch ein ETL-Tool zunächst aufbereitet werden.	Input Prozess: https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/upload-data/
DI5	Metadaten, welche in den Datenquellen enthalten sind sollen mit ausgelesen werden.	o	Möglich, dafür müssen die Metadaten aber in einem zu implementierenden Präprozess, für den die .Stat Suite keine Werkzeuge bietet, in SDMX-Strukturdefinitionen übertragen werden (Zusatzleistung nötig).	
DI6	Während des Data-Inputs sollen zusätzliche Metadaten für einen Datensatz angereichert werden können.	o	siehe vorherige Zeile	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DI7	Während des Data-Inputs sollen Verlinkungen zu anderen Datensätzen im Data-Store möglich sein.	o	Eine direkte Verlinkung ist nicht möglich. Eine Verbindung zwischen Datensätzen herzustellen, kann aber über geteilte Metadaten (= Elemente der SDMX-Strukturdefinitionen) hergestellt werden, indem dieselben Dimensionen wie für andere Datensätze verwendet werden.	
DI8	Mit dem Data-Input soll es möglich sein, mehrere Versionen von einem Datensatz einzulesen und im Data-Store zu speichern.	+	SDMX Elemente können mit verschiedenen Versionsnummern versehen werden. Dieser Schritt muss vor dem Import durch die SDMX-Datei geschehen.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/dlm_overview/
DI9	Für Datensätze mit gleichbleibenden Strukturen soll ein automatisierter Importprozess möglich sein.	o	Ein solcher Importprozess kann über die Nutzung der bereitgestellten SDMX-APIs umgesetzt werden. Die automatisierte Kommunikation mit der SDMX-API könnte durch ein ETL-Tool implementiert werden. In der .Stat Suite ist hierfür kein Werkzeug enthalten.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-api/api-main-features/
DI10	Für Automatisierungen soll es möglich sein, zeitliche Ablaufpläne zu konfigurieren, um Datensätze regelmäßig zu importieren.	o	Abhängig von ETL-Tool (siehe DI9), in .Stat Suite nicht vorhanden.	
RP1	Es muss ein redaktioneller Prozess zur Pflege der Daten zur Umwelt entwickelt werden, oder der DataCube muss in den bestehenden Prozess mit eingebettet werden.	o	Angesichts der Notwendigkeit von Präprozessen (siehe Anforderungen in DS) ist die Einbindung der .Stat Suite in einen umfassenden Workflow in jedem Fall erforderlich. Die Werkzeuge für diesen Workflow müssen neu gebaut oder mithilfe einer externen Softwarekomponente implementiert werden, die	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			.Stat Suite enthält nicht die nötigen Werkzeuge hierzu.	
RP2	Daten dürfen nur nach einer Überprüfung durch einen Data-Output auf der Webseite dargestellt werden.	+	Durch Datenhaltung in verschiedenen Data Spaces, für die je unterschiedliche Rechteverwaltung möglich ist, umsetzbar (z.B. Data Space „in Bearbeitung“ und Data Space „öffentlich“). Die .Stat Suite bietet allerdings keine Werkzeuge oder Kontrollstrukturen, um derartige Workflows umzusetzen. Dies müsste organisatorisch oder durch zusätzliche Software geschehen.	https://sis-cc.gitlab.io/dotstatsuite-documentation/about/product-overview/
RP3	Data-Outputs müssen aufwandsarm in Artikel der UBA Webseite eingebettet werden können.	+	Als „embedded code“ und als URL share-service möglich (siehe DO1)	
RP4	Data-Outputs dürfen nur nach vorheriger Freigabe veröffentlicht werden.	o	Abhängig von der Implementierung (s. RP2)	
RP5	Bei Anpassung der Daten eines Berichts/Artikels sollen Redakteure benachrichtigt werden.	o	Unabhängig von .Stat Suite. Muss im Data-Input implementiert werden.	
RP6	Der manuelle Aufwand des redaktionellen Prozesses soll soweit möglich reduziert werden.	o	Die .Stat Suite ist an vielen Stellen auf manuelle Eingabe angewiesen (z.B. im DLM beim Upload von Daten und Metadaten). Je nach Implementierung (s. DI9) könnten einzelne Schritte über die SDMX-API automatisiert werden.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/upload-structure/
DS1	Es ist eine zentrale Datenhaltung mit allen Daten zur	+	Umfassende Datenhaltung auf Basis von (SDMX-) Strukturinformationen. Es ist möglich bzw. vorgesehen,	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Umwelt aufzubauen.		mehrere voneinander getrennte Datenhaltungen (sog. „Data Space“) für unterschiedliche Stadien der Reife der Daten zu verwenden (z.B. ein „in Bearbeitung“-Data Space für Import und Qualitätskontrolle und ein Data Space „finalisiert“ für die Daten, die alle Kontrollen bestanden haben und öffentlich zugänglich sind). Jeder Data Space besteht aus zwei Komponenten, einem für die SDMX-Strukturinformationen (die Metadaten) und eine für die Nutzdaten selbst. Die einzelnen Data Spaces sind je einzeln über eine SDMX-API erreichbar.	
DS2	Die Datenstrukturen müssen klar definiert werden. Es muss ein themenübergreifender Gesamtansatz für das Datenmodell ausgearbeitet werden, in den sich die verschiedenartigen Daten zur Umwelt einordnen.	+	Kann durch SDMX-Strukturdefinitionen umgesetzt werden. Alle im System zu verwaltenden Daten sind durch SDMX-Strukturdefinitionen zu beschreiben. Die Definitionen der einzelnen Datensätze können auf geteilte Strukturelemente (u.a. Codelists, Dimensionen, Konzepte) zugreifen. Auf dieser Grundlage kann ein umfassendes Metadatensystem aufgebaut werden, das die einzelnen Datensätze über gemeinsame Strukturelemente verbindet.	
DS3	Das Datenmodell muss so konzipiert werden, dass es einerseits beliebig vertieft/ detailliert und andererseits beliebig fachlich erweitert werden kann.	o	Neue Strukturdefinitionen können hinzugefügt, vorhandene aktualisiert werden. Eine Erweiterung ist auf diese Weise leicht möglich. Eine Vertiefung im Sinne einer objektorientierten Vererbung ist nicht möglich.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DS4	Die datenhaltenden Stellen sollen ihre Daten möglichst automatisiert in den Data-Store einspielen können.	o	Der Daten-Import ist über den Upload von SDMX-Dateien realisiert. Falls die Daten nicht in SDMX vorliegen, sind ETL-Prozesse zur Konvertierung notwendig. Alternativ zum manuellen Upload könnten die APIs verwendet werden (s. DI9).	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/upload-data/
DS5	Auch rückwirkend müssen Datenaktualisierungen und -erfassungen im Data-Store an die Fachsysteme (teil-) automatisiert übergeben werden können.	+	Im DLM können Datensätze jederzeit gelöscht und anschließend neu hochgeladen werden. Alternativ können Datensätze unter einer neuen Versionsnummer zur Verfügung gestellt werden.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/delete-data-structures/ https://sis-cc.gitlab.io/dotstatsuite-documentation/using-api/api-main-features/
DS6	Die Webseite soll Daten direkt aus dem Data-Store nutzen.	+	Der DE und die Datenhaltung sind direkt verknüpft. Alternativ können SDMX-APIs genutzt werden, um eine externe Webseite mit aktuellen Daten aus der Datenbank zu versorgen.	https://sis-cc.gitlab.io/dotstatsuite-documentation/getting-started/framework/#stat-data-explorer-module
DS7	Datenänderungen sollen automatisch an die Redakteure bzw. Fachexperten/-innen übermittelt werden.	+	Es ist ein E-Mail-Benachrichtigungssystem vorhanden, das Nutzende über Änderungen der Daten im DLM und über die APIs informiert, die sie vorgenommen haben.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-api/message-through-mail/
DS8	Die Daten müssen über Status so markiert sein, dass sofort erkennbar ist, in welchem Bearbeitungszustand sie sich gerade befinden.	o	Ein solches Statuskonzept gibt es nicht. Es ist nur möglich, unterschiedliche Data Spaces zu unterhalten, die diese Status ausdrücken (siehe „DS1 Datenhaltung“).	
DS9	Dynamische Zugriffe auf die Daten müssen in Abhängigkeit des Status definiert und kontrolliert werden können.	o	Siehe vorherige Zeile	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DS10	Es muss möglich sein, Daten aus verschiedenen Quellsystemen (insbesondere aus DESTATIS) zu übernehmen, diese einheitlich strukturiert zu verwalten, sodass sie dann als Dimensionen der unterschiedlichen Daten zur Umwelt im Data-Store genutzt werden können.	o	Aufgrund der SDMX-basierten Datenhaltung ist die Verwaltung und Nutzung sehr gut umsetzbar. Aufgrund der Einschränkungen bei der Übernahme der Daten (s. DS4) müsste das Auffinden gemeinsamer Dimensionen und das Transformieren der neuen Daten in SDMX als ein Präprozess definiert und umgesetzt werden. Die .Stat Suite kann diesen Präprozess nicht leisten. (Zusatzleistung nötig)	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-api/core-data-model
DS11	Es muss möglich sein, für alle erfassten Datensätze Dimensionen zu definieren.	+	Mit SDMX sehr gut möglich. Allerdings müssen die Datensätze zuerst (in einer SDMX-Strukturdefinition) beschreiben werden, bevor sie im System erfasst werden können. Das System bietet keine Werkzeuge für die Strukturdefinition an.	https://sdmx.org/?page_id=2555 https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/edit-structure/
DS12	Es muss möglich sein, die übernommenen Daten (konkrete Werte der Datensätze) den entsprechenden Dimensionen zuzuordnen.	+	Mit SDMX sehr gut möglich. Das Format ist genau dazu gedacht.	
DS13	Die Werte müssen auch als Werte (Zahlen, Vektoren) gespeichert werden, sodass mit ihnen gerechnet werden kann.	+	Die Speicherung geschieht ausschließlich als Werte.	
DS14	Es soll möglich sein, im Data-Store Berechnungen zu hinterlegen, die automatisch	o	Keine Werkzeuge dafür vorhanden. Berechnungen könnten grundsätzlich auch im Data-Input durchgeführt werden.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	ausgeführt werden.			
DS15	Es soll möglich sein, bestimmte Berechnungen manuell anzustoßen.	-	Keine Werkzeuge dafür vorhanden.	
DS16	Es soll möglich sein, Berechnungen hierarchisch auszuführen. Das gilt auch für automatisierte Berechnungen.	-	Keine Werkzeuge dafür vorhanden.	
DS17	Es muss auch weiterhin möglich sein, extern mit den Daten zur Umwelt zu arbeiten und externe Daten mit ihren Bezügen zu den Ausgangsdaten im Data-Store einzuspielen.	+	Es existiert ein Excel-Addin um bereits in der .Stat Suite existierende Daten zu bearbeiten und entsprechende Änderungen zurückzuspielen.	https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm-excel-addin/
DS18	Die Werte müssen historisch verwaltet werden.	-	Historisierung wird nicht unterstützt.	
DS19	Bei hierarchischen Berechnungen müssen die Bezüge der Daten jederzeit wieder selektierbar sein.	-	Nicht möglich.	
DS20	Objektklassen, die für die Dimensionierung herangezogen werden, müssen ebenfalls historisch verwaltet werden, um ältere Daten zur Umwelt auch	-	Nicht möglich.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	weiterhin einordnen zu können.			

3.5.1.3 Bewertung

Im Folgenden sind zunächst die Stärken der .Stat Suite beschrieben. Anschließend wird auf besondere Herausforderungen eingegangen, die für den Einsatz dieser Lösungsalternative gesehen werden.

Die .Stat Suite basiert auf dem Format SDMX, welches für die Bereitstellung und Beschreibung multidimensionaler Daten gut geeignet ist. Der open-source-Ansatz ermöglicht es, Komponenten anzupassen und ihre interne Funktionsweise recherchieren zu können. Eine zusätzliche Flexibilität gewinnt die .Stat Suite durch die lose Kopplung der drei Komponenten über Dienste. Die Dienste, insbesondere die SDMX-APIs, sind gute Ausgangspunkte für eine mögliche Automatisierung und einen flexiblen Datenfluss, um z. B. externe Webseiten mit Inhalten zu bedienen oder einen umfassenden Workflow mit den .Stat Suite-Komponenten und weiteren Softwarekomponenten zu gestalten. Üblicherweise führen Anpassungen an open-source Lösungen jedoch zu Schwierigkeiten bei der langfristigen Pflege zum Beispiel beim Einspielen von Updates.

Der Data-Input ist in der .Stat Suite sehr unflexibel und restriktiv. Das ganze System basiert darauf, vorbereitete SDMX-Dokumente, die durch korrekte und vollständige Metadaten beschrieben sind, zu verarbeiten. Es bietet keine Werkzeuge, um diese Dokumente oder die benötigten Metadaten herzustellen. Die .Stat Suite ist vor allem in einem Workflow sinnvoll, in dem die Daten bereits in SDMX formatiert geliefert werden. Sonst ist der Einsatz der .Stat Suite abhängig von einem Präprozess, der die geschilderten Aufgaben übernimmt. Um eine kohärente Beschreibung mit Metadaten zu erreichen, muss dieser Präprozess sehr eng mit der Datenhaltung vernetzt werden.

Die drei dargestellten Komponenten der .Stat Suite bestehen jeweils aus mehreren einzelnen Datenbanken und Diensten. So besteht beispielswiese der .Stat Core aus einem Transferdienst, einem SDMX-Dienst und einem Autorisierungsdienst, die auf eine Datenbank für Nutzdaten (“data-db"), eine für Metadaten (“structure-db") und eine für globale Einstellungen (“common”) zugreifen (Statistical Information System Collaboration Community - SIS-CC, o. J. - e). Ein ähnliches Bild bietet die auf der .Stat Suite-Seite publizierte Demoversion (Statistical Information System Collaboration Community - SIS-CC, o. J. - v). Wird sie heruntergeladen (z. B. aus dem zugehörigen git repository (Statistical Information System Collaboration Community - SIS-CC, o. J. - b)) und im Containerprogramm Docker Compose gestartet, sind dort 21 voneinander getrennte Prozesse zu sehen (unter anderem Authentifizierungsdienste, Datenbanken, Webserver, API-Server). Diese Dienste und Datenbanken basieren auf etablierten und weitverbreiteten Technologien (z. B. .Net von Microsoft). Dennoch sind aufgrund der Menge an Diensten und Datenbanken, deren Betrieb und deren Kommunikation untereinander gewährleistet sein muss, bei Installation, Konfiguration und Wartung besondere Herausforderungen und gegebenenfalls resultierende Verzögerungen zu erwarten. Insbesondere zukünftige Updates des Gesamtsystems oder einzelner Teile bei laufendem Betrieb sind kritisch zu bewerten. Eine Installationsroutine für den on-premise Betrieb existiert dabei nicht. Alle Komponenten müssen einzeln in Betrieb genommen werden. Eine technische

Unterstützung ist nur für Mitglieder der .Stat Community möglich, welche durch regelmäßige Mitgliedsbeiträge finanziert wird. Die Höhe der Beiträge ist über die Webseite nicht bekannt (Statistical Information System Collaboration Community - SIS-CC, o. J. - ae).

Vor allem der .Stat Data-Explorer bietet gute Möglichkeiten zur Exploration von Daten, da Nutzende eigenständig Daten suchen und in frei wählbaren Darstellungsarten visualisieren können. Data-Outputs zur Integration in das UBA-CMS existieren jedoch nicht und müssten als eigene Drupal-Module implementiert werden. Hierzu kann voraussichtlich die SDMX-API zur Anbindung an die Datenhaltung verwendet werden. Alle Funktionalitäten zur Exploration, Darstellung und Interaktion müssen jedoch implementiert werden. Eine Integration des .Stat Data-Explorer ohne iFrames ist nicht angedacht.

Als positiv zu bewerten ist, dass für die Nutzung der .Stat Suite keine Lizenzkosten anfallen. Bei der Komponente des Microsoft SQL Server sind jedoch verschiedene Lizenzstufen möglich. Hier kann gegebenenfalls auf die bereits existierenden Microsoft SQL Server des UBA zurückgegriffen werden.

3.5.2 Highcharts

3.5.2.1 Lösungsansatz

Im Vergleich zu den weiteren vorgestellten Lösungskomponenten stellt Highcharts lediglich eine JavaScript Bibliothek zur Implementierung eigener interaktiver Diagramme zur Verfügung. Diese bieten eine sehr gute Grundlage für Entwicklungen, da viele der interaktiven Anforderungen durch Highcharts abgebildet werden können. Es handelt sich jedoch nicht um eine fertige, installierbare Anwendung, was bedeutet, dass alle Komponenten neuentwickelt werden müssen. Dabei würde die Highcharts-Bibliothek als Basis für die Abdeckung bestimmter Anforderungen an die Data-Outputs oder den Data-Explorer dienen. Im Folgenden soll daher ein Lösungsansatz zur Entwicklung eigener Komponenten beschrieben werden.

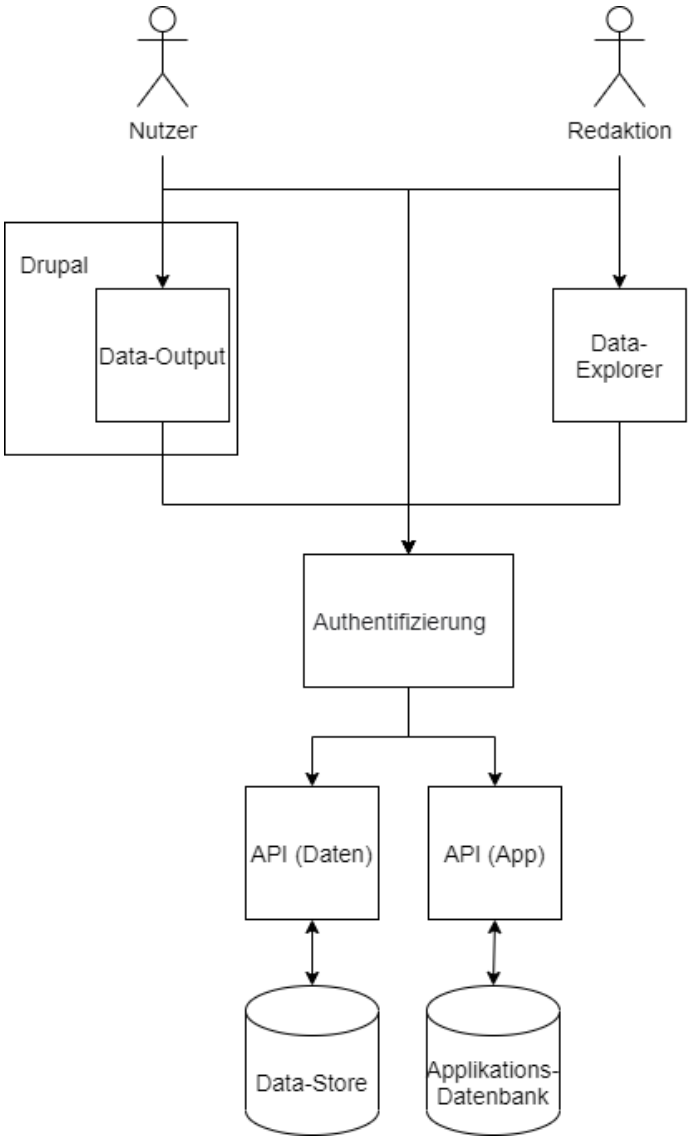
In Abbildung 6 sind die notwendigen Komponenten für eine Eigenentwicklung und deren Zusammenhänge dargestellt: Zusätzlich zu der Data-Store Komponente muss eine Datenhaltung für anwendungsbezogene Daten aufgebaut werden. Diese wird im Folgenden mit Applikations-DB beschrieben und soll zum Beispiel dafür genutzt werden, dass Data-Output Konfigurationen gespeichert werden können. Inhalte des Data-Stores als auch der Applikations-DB müssen für Data-Explorer und Data-Output zugreifbar sein, wofür jeweils eine API bereitgestellt werden muss. Diese sind in der Abbildung mit API (Daten) und API (App) bezeichnet und müssen als Server-Anwendung implementiert werden.

Die Data-Store API muss Abfragen für Daten ermöglichen, die sowohl für externe Nutzende als auch durch Data-Explorer und Data-Output genutzt werden können. Die Anforderungen durch die Nutzenden der API (z. B. zur Anbindung in Python-Skripten oder ähnlichem) ist zu großen Teilen deckungsgleich mit den Anforderungen durch Data-Outputs: Ein Zugriff auf Daten mit verschiedenen Filter-Möglichkeiten müsste ebenso wie eine Export-Funktionalität implementiert werden. Da nicht alle Daten öffentlich sein werden, muss zusätzlich eine Komponente zur Absicherung der Anfragen implementiert werden. Hier könnten open-source Lösungen wie zum Beispiel Keycloak (Keycloak Authors, The Linux Foundation, o. J.) für die Authentifizierung verwendet werden.

In der Applikations-API müssen Data-Output Konfigurationen gelesen und geschrieben werden können. Auch hier ist eine Authentifizierung notwendig, da nur Nutzende der Redaktion diese Konfigurationen anpassen bzw. erstellen dürfen.

Data-Outputs und Data-Explorer können als eigenständige Webseiten implementiert werden, wobei für die Data-Outputs zusätzlich eine Integration in eigene Drupal Module erfolgen muss. Die grundlegende Data-Output Funktionalität müsste dabei möglichst CMS neutral erfolgen, um auch eine Einbettung in umwelt.info zu ermöglichen. Das Design der Komponenten ist komplett frei zu gestalten.

Abbildung 6: Architektur für eine Entwicklung auf Basis von Highcharts



Quelle: eigene Darstellung, con terra GmbH

Highcharts selbst bietet eine große Anzahl von Möglichkeiten zur Visualisierung von interaktiven Diagrammen. Es sind viele verschiedene Diagramm-Typen möglich (Highsoft AS, o. J. - a), und zusätzlich sind Basis-Funktionen für Drilldowns (Deaggregationen) (Highsoft AS, o. J. - b), Labels (Highsoft AS, o. J. - f) und Zoom (Highsoft AS, o. J. - e) vorhanden, welche beliebig weiter ausgebaut werden können. Durch die freie Implementierung sind viele Möglichkeiten der Exploration (Hinzuladen von weiteren Datensätzen, Anpassung der Konfiguration durch Nutzende usw.) möglich, die in den bereits existierenden Anwendungen fehlen. Zur Konfiguration von Diagrammen ist bereits eine open-source Komponente mit dem Namen Highcharts-Editor verfügbar (Highsoft AS, o. J. - d). Diese könnte als Grundlage für einige Funktionen im Data-Output und Data-Explorer verwendet werden.

Die Wahl der konkreten Datenhaltung ist unabhängig von der Implementierung auf Basis von Highcharts. Daher sind in diesem Zusammenhang beliebige Lösungskomponenten denkbar, die die Anforderungen an den Data-Store abbilden. In diesem Sinne ist die Data-Input Komponente ebenfalls losgelöst von Highcharts zu betrachten. Grundsätzlich können die API- und GUI-Komponenten jedoch frei implementiert werden, wodurch diese auch ergänzende Funktionalitäten bereitstellen können. Je nach Wahl des Datenmodells im Data-Store könnte die API-Komponente z. B. auch verwendet werden, um Datenbank-Sichten zu erzeugen und so z. B. Datensätze neu als Cubes zusammenzustellen.

3.5.2.2 Anforderungstabelle

In der folgenden Tabelle werden alle Anforderungen an den Data Cube durch die Lösungskomponente beschrieben und bewertet. Da in diesem Kapitel größtenteils eine Eigenentwicklung beschrieben wird ist die Bewertung abweichend zu den anderen Lösungskomponenten zu sehen. Durch (+) werden Anforderungen beschrieben, die einfach umsetzbar sind, oder für die durch Highcharts bereits entsprechende Basis-Funktionalitäten bereitstehen. (o) beschreibt eine mit vertretbarem Aufwand umsetzbare Anforderung, und durch (-) werden nur mit hohem Aufwand implementierbare Anforderungen bewertet.

Tabelle 3: Anforderungstabelle der Lösungskomponente Highcharts

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
AA1	Die Software soll erweiterbar/ anpassbar sein, falls nicht alle Anforderungen abgedeckt werden können (Anpassung von Design, aber auch Funktionalitäten)	+	Highcharts ist eine JavaScript Bibliothek, die lediglich die Grundfunktionalitäten zur Erstellung von interaktiven Diagrammen bereitstellt. Die einzelnen Anwendungen müssen auf Basis der Bibliothek entwickelt werden. Die Funktionalitäten sind damit sehr anpassbar.	
AA2	Die Anforderungen müssen durch die Software effizient implementiert werden können.	-	Alle Komponenten müssen neu implementiert werden. Der initiale Aufwand ist daher sehr hoch.	
AA3	Die Software soll mit geringem Aufwand in Betrieb genommen werden.	o	Die Bibliothek selbst muss nur in Webseiten eingebettet werden. Die einzelnen Komponenten (Explorer, Output) müssten jedoch individuell bereitgestellt werden.	
AA4	Die Software muss für Anwender und Administratoren ausreichend	+	Es ist eine sehr gute Entwicklerdokumentation verfügbar.	https://www.highcharts.com/docs/index

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	dokumentiert sein.			
AA5	Die Software ist bereits ausgereift und kann direkt verwendet werden.	o	Highcharts ist eine etablierte Bibliothek mit vielen Nutzenden. Viele Komponenten wären jedoch neue Entwicklungen.	
AA6	Es gibt eine aktive Community mit Diskussionsforen und Beispielen zur Anwendung.	+	Eigene Community Seite mit Forum	https://www.highcharts.com/forum/
AA7	Die Software ist kostenfrei nutzbar, oder die Kosten sind in einem für das Projekt vertretbaren Rahmen.	o	Highcharts muss lizenziert werden. Eine Lizenz ist im UBA allerdings bereits vorhanden. Die verbleibenden Komponenten müssen jedoch entwickelt werden.	https://www.highcharts.com/download/
AA8	Die Software kann im UBA Rechenzentrum betrieben werden	+	Die Software kann im UBA Rechenzentrum installiert werden (on-premise Lösung).	
AA9	Die Software kann ohne Weiteres gratis getestet werden	+	Highcharts kann gratis getestet werden.	
DO1	Es muss eine Visualisierung von Daten möglich sein, bei denen eine feste Ansicht auf die Daten und eine feste Darstellungsform durch die Redaktion vorgegeben wird.	o	Highcharts Diagramme können durch eine JSON Konfiguration definiert werden. Zu implementieren: Für eine feste Darstellung müssen alle Parameter (Datenquelle, Darstellungsoptionen) im Data-Explorer definiert und entsprechend abgespeichert werden. Der konkrete Data-Output muss initial die JSON-Konfiguration über die API laden und entsprechend anwenden.	
DO2	Daten müssen als Tabelle dargestellt werden können.	o	Highcharts selbst bietet keine interaktiven Tabellen. Diese können jedoch durch eine zusätzliche Bibliothek ergänzt werden (z.B. Vuetify). Zu implementieren: Die entsprechenden Daten müssen	https://vuetifyjs.com/en/components/data-tables/

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			vorher über die API abgerufen werden. Die Konfiguration der Tabellen-Ansicht müsste im Data-Explorer erfolgen und entsprechend für den Data-Output gespeichert werden.	
DO3	Daten innerhalb der Tabellendarstellung sollen gefiltert werden können.	+	In der zuvor genannten Bibliothek können Filter implementiert werden.	
DO4	Daten innerhalb der Tabellendarstellung sollen sortiert werden können.	+	In der zuvor genannten Bibliothek können Spalten im Standard sortiert werden.	
DO5	Aggregationen von Datensätzen sollen vorgenommen oder vorberechnete Aggregationen angezeigt werden können.	o	Vorberechnete Aggregationen können wie "normale" Datensätze dargestellt werden und könnten als verlinkte Datensätze innerhalb der Cubes angeboten werden. Dynamische Aggregationen müssten als zusätzliche Funktion für die Data-Outputs implementiert werden. Hierzu müsste zunächst der Datensatz über die API geladen werden. Anschließen könnte der Nutzende das Attribut zur Gruppierung sowie die Aggregations-Funktion (Summe, Minimum, Maximum, Mittelwert) über Dropdown-Menüs auswählen. Die Aggregation selbst könnte im JavaScript Code durchgeführt werden.	
DO6	Deaggregationen von Datensätzen sollen vorgenommen oder vorberechnete Deaggregationen angezeigt werden können.	-	Vorberechnete Deaggregationen können wie "normale" Datensätze dargestellt werden und könnten als verlinkte Datensätze innerhalb der Cubes angeboten werden. Für dynamische Deaggregationen kann die Drilldown-Funktion von Highcharts verwendet werden. Der Drilldown kann asynchron erfolgen. Das bedeutet, dass Daten innerhalb des Diagramms	https://www.highcharts.com/docs/chart-concepts/drilldown

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			nach einem Klick durch detailliertere Daten ausgetauscht werden können. Die neuen Daten müssten daher auch über die API geladen werden und daher im Voraus als Hierarchie definiert sein. Highcharts bietet hier die Basisfunktionalität zum Identifizieren der Drilldown-Klicks, das konkrete Nachladen der Daten muss implementiert werden.	
DO7	Zeitreihen sollen eingeschränkt werden können (Filterung durch min. und max. Datum).	-	Ein Datum-basierter Filter existiert im Standard nur für Stock-Charts. Für alle anderen Diagramme muss diese Funktion implementiert werden. Das Filtern könnte sowohl im JavaScript Code als auch durch eine neue API-Abfrage erfolgen. Die entsprechenden Daten müssen dazu zeitlich gefiltert und im Diagramm ersetzt werden.	
DO8	Einheiten sollen dynamisch umgerechnet werden, oder durch vordefinierte Daten in anderen Einheiten angezeigt werden können.	o	Zu implementieren: Vorberechnete Datensätze könnten als verlinkte Datensätze angeboten werden. Für dynamische Umrechnungen müsste der entsprechende Faktor für die Dimension abgelegt und über die API bereitgestellt werden. Die einzelnen Werte könnten im JavaScript Code direkt umgerechnet und im Diagramm aktualisiert werden.	
DO9	Die Darstellungsform der Abbildungen soll durch den Nutzenden frei gewählt werden können. Gewünschte Darstellungsformen sind: Liniendiagramm, Balkendiagramm, Tortendiagramm,	o	Für eine Anpassung der Darstellung könnte der Highcharts-Editor mit eingebunden werden. Der Highcharts-Editor ist open-source und kann frei in andere Anwendungen integriert werden. Eine Anbindung an eine Datenbank oder API ist noch nicht vorhanden.	https://www.highcharts.com/products/highcharts-editor/

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Baumdiagramme, Abweichungen, Korrelationen und Streudiagramme, Häufigkeitsverteilungen, Nominale Vergleiche (wie Blasendiagramme oder Heatmaps), Fluss und Sankey-Diagramme, Netzwerkdiagramme)			
DO10	Die Visualisierungen in Diagrammen/Abbildungen (Data-Outputs) müssen direkt in Artikel auf der UBA-Webseite (Drupal CMS) eingebunden werden können	o	Es kann ein Drupal Modul implementiert werden, um eine native Integration zu ermöglichen. Durch eine API-Anbindung könnten Daten und Data-Output Einstellungen direkt aus der Datenhaltung verwendet werden. Mit diesen Informationen kann anschließend die GUI dynamisch für den jeweiligen Data-Output generiert werden. Innerhalb des Modules muss die Highcharts-Bibliothek integriert werden.	
DO11	Die Visualisierungen in Diagrammen/Abbildungen (Data-Outputs) soll in der zukünftigen umwelt.info Webseite möglich sein.	o	Eine Einbettung in die umwelt.info Webseite erfolgt analog zur Einbettung in Drupal.	
DO12	Die Implementierung soll möglichst CMS-offen (unabhängig von dem bestehenden Drupal System) erfolgen. Es soll geprüft werden ob Inhalte z.B. als reponsive iFrames oder vergleichbares	+	Implementierung als reine HTML/JavaScript Anwendung möglich.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	eingebettet werden können.			
DO13	Funktionalitäten die nicht CMS-offen implementiert werden, sind als Drupal-Modul (unter Beachtung der Drupal Code-Konventionen) zu entwickeln in vollständig lauffähig in das CMS der UBA Webseite zu integrieren.	+	Code-Konventionen können bei der Entwicklung eigener Module beachtet werden.	
DO14	Es muss möglich sein, Dashboards im Drupal CMS einzubinden, die aus mehreren Data-Outputs bestehen	o	Dashboards bestehen aus verschiedenen Data-Outputs. Es muss ein Tool zum Anordnen verschiedener Data-Outputs implementiert werden. Die Einbettung kann anschließend analog zu einzelnen Data-Outputs erfolgen (DE17)	
DO15	Dashboards und Data-Outputs sollen auf verschiedenen Display Größen (Mobilgeräte und größere Bildschirme) gut angezeigt werden.	+	Highcharts selbst ist responsiv. Die weiteren Funktionalitäten (Dashboards, UI-Elemente) müssten ebenfalls responsiv implementiert werden.	
DO16	Daten und Abbildungen sollen in unterschiedlichen Formaten zum Download angeboten werden. Daten: CSV, Excel Abbildungen: PNG/JPEG	+	Highcharts bietet bereits Exporte für CSV, Excel, HTML sowie JPG, PNG, SVG, und PDF. Weitere Exporte können über eigene Export-Erweiterungen implementiert werden. Je nach Format könnte FME Server für Konvertierungen angebunden werden. Die Highcharts Exporte funktionieren nur für die aktuell im Diagramm angezeigten Daten.	https://www.highcharts.com/docs/export-module/export-module-overview https://www.highcharts.com/docs/export-module/client-side-export
DO17	Datensätze die zu einem Thema gehören sollen	o	Funktionalität muss neu implementiert werden. Über die API müssten verwandte	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	erkundet werden können. Das heißt, dass die Daten sowie verlinkte Datensätze angezeigt werden.		Datensätze aufgelistet und angezeigt werden. Die Funktionalität kann entweder innerhalb der Data-Outputs oder als eigenständige Komponente implementiert werden.	
DO18	Verknüpfte Datensätze sollen in der Visualisierung mit hinzugefügt werden können, um Daten vergleichen zu können.	o	In Highcharts können verschiedene Datensätze gleichzeitig visualisiert werden. Das Hinzufügen von Daten aus dem Data-Store zu einem bestehenden Data-Output muss jedoch implementiert werden. Die Funktionalität benötigt DO17 zur Auflistung weiterer Datensätze. Nach der Selektion könnten diese dem aktuellen Data-Output hinzugefügt werden.	https://www.highcharts.com/docs/chart-and-series-types/combining-chart-types
DO19	Das Corporate Design ist bei der Gestaltung aller Nutzeroberflächen zu beachten.	+	Kann bei der Implementierung beachtet werden.	
DO20	Es ist zu berücksichtigen wie Data-Outputs auch in die Datensuche des UBA eingebunden werden können.	o	Wenn Data-Outputs als Drupal-Module implementiert werden, kann der Standard Drupal-Mechanismus verwendet werden. Hierzu müssen beim Erstellen der Inhalte in Drupal entsprechende Keywords vergeben werden.	
DE1	Über den Data-Explorer müssen Datensätze des Data Stores auffindbar gemacht werden.	o	Der Data-Explorer muss neu implementiert werden. Es wird eine API benötigt, die alle Datensätze auflistet. Der Data-Explorer muss anschließend alle Datensätze auflisten und entsprechend darstellen.	
DE2	Metadaten von Datensätzen sollen angezeigt werden können.	o	Nach der Selektion eines Datensatzes im Data-Explorer (DE1) müssen über eine API entsprechende Metadaten nachgeladen und angezeigt werden. Diese Funktionalität muss neu implementiert werden.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DE3	Datensätze sollen als interaktive Diagramme angezeigt werden können.	+	Die Funktionen des Data-Output können hier verwendet werden und müssten im Data-Explorer ebenfalls integriert werden.	
DE4	Datensätze sollen tabellarisch angezeigt werden können.	+	siehe DE3	
DE5	Eine Definition von zusammengehöri gen Datensätzen für einen Cube soll möglich sein.	-	Es muss möglich sein, alle Datensätze aufzulisten (DE1) und über die API auch neue Cubes zu erstellen. Dazu müsste zunächst ein neuer Cube über die API in der Datenbank angelegt werden. Anschließend müssten beliebig viele Datensätze über die GUI ausgewählt und hinzugefügt werden können. Diese Funktionalität muss neu implementiert werden.	
DE6	Es soll möglich sein, alle Cubes aufzulisten.	o	Analog zu DE1 müssen auch Cubes aufgelistet werden können.	
DE7	Es soll möglich sein alle Datensätze aufzulisten, die in einem Cube verwendet werden.	o	Analog zu DE1 müssen auch Verlinkungen zwischen Datensätzen und Cubes angezeigt werden können. Hierzu ist eine entsprechende API-Funktionalität notwendig.	
DE8	Es soll möglich sein für einen Datensatz anzuzeigen, in welchem Cube er verwendet wird.	o	Analog zu DE1 müssen auch Verlinkungen zwischen Datensätzen und Cubes angezeigt werden können. Hierzu ist eine entsprechende API-Funktionalität notwendig.	
DE9	Für jeden Datensatz sollen fachlich begründete Dimensionen ausgewählt werden können.	-	Beim Hinzufügen von Datensätzen zu Cubes (DE5) müssten Filterfunktionen implementiert werden, um die ausgewählten Datensätze weiter auf ihre Dimension beschränken zu können. Diese Filterung muss für den Cube gespeichert werden und bei allen zukünftigen Abfragen berücksichtigt werden. Hierzu	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			muss sowohl eine API als auch die entsprechende GUI implementiert werden.	
DE10	Hierarchische Datensätze sollen miteinander verknüpft werden können.	-	Für jeden Datensatz muss über die GUI eine Verknüpfung als Kind- oder Elternobjekt zu einem anderen Datensatz hergestellt werden können. Die Information muss anschließend im Cube gespeichert werden. Hierzu muss sowohl eine API als auch die entsprechende GUI implementiert werden.	
DE11	Datenauszügen für einzelne Datensätze sollen definiert werden können.	-	Ähnlich zu DE9 müssen entsprechende Filter definierbar sein, die für zukünftige Abfragen berücksichtigt werden. Auszüge aus Datensätzen sollten nach der Erstellung wie eigenständige Datensätze nutzbar sein. Hierzu muss sowohl eine API als auch die entsprechende GUI implementiert werden.	
DE12	Eine Konfiguration von Datensätzen / Dimensionen für einzelne Data-Outputs soll ermöglicht werden. Es soll konfigurierbar sein, welche weiteren Datensätze zu einem Data-Output potenziell hinzugeladen werden dürfen.	o	Durch DE9, DE10, DE11 sind bereits gefilterte Datensätze vorhanden. Diese müssen ähnlich wie bei der Cube Zusammenstellung (DE5) auch für einzelne Data-Outputs möglich sein.	
DE13	Konfiguration der initialen und konfigurierbaren Darstellungsform	o	Ähnlich zu DO9 kann hier der Highcharts-Editor als Basis verwendet werden. Der Editor muss jedoch für die Verwendung des Data-Store angepasst werden. Anstelle des Datei-Uploads im Web, müssten Daten direkt über eine API ausgelesen werden. Die Konfiguration müsste	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			anschließend für den jeweiligen Data-Output in der Datenbank gespeichert werden.	
DE14	Konfiguration weiterer Optionen des Data-Outputs für die Nutzenden: - Daten hinzuladen - Einheiten umrechnen / Datensatz mit anderen Einheiten laden - Aggregationen berechnen / Datensätze mit anderen Aggregationen laden - Erlaubte Downloads	o	Die Konfigurationen in DE12 und DE13 müssen in der Datenbank gespeichert werden. Als Basis für die GUI Entwicklungen könnte der Highcharts-Editor verwendet werden.	
DE15	Für jeden Data-Output sollen die Zusatz-Funktionalitäten (z.B. Aggregationen berechnen, Daten hinzuladen, …) bei Bedarf deaktiviert werden können.	o	siehe DE14	
DE16	Konfiguration von Dashboards (Kombination aus mehreren Data-Outputs)	o	Dashboards bestehen aus verschiedenen Data-Outputs. Es muss ein Tool zum Anordnen verschiedener Data-Outputs implementiert werden. Die Einbettung kann anschließend analog zu einzelnen Data-Outputs erfolgen.	
DE17	Der Data-Explorer soll außerhalb des UBA CMS implementiert werden können und hat kein spezifisches CMS zur Voraussetzung.	+	Der Data-Explorer ist als eigenständige Web-Anwendung gedacht.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DE18	Für externe Analysetools soll der Zugriff auf die Daten über eine API ermöglicht werden.	+	Zur Implementierung der Data-Explorer und Data-Output Komponenten muss eine eigene API entwickelt werden. Diese kann ebenfalls für andere Anwendungen verwendet werden.	
DE19	Für externe Analysetools soll der Zugriff auf die Daten über einen Export ermöglicht werden.			
DE20	Die Konfiguration der Data-Outputs und Data Cubes soll über ein Webinterface möglich sein.	+	Die zuvor beschriebenen Entwicklungen sind als Web-Anwendung konzipiert.	
DE21	Es soll möglich sein Zusatzdateien für Berichte (Bilder in verschiedenen Formaten, Daten) herunterzuladen.	o	Exporte können über die Highcharts Diagramme durchgeführt werden (DO16). Falls weitere Exportmöglichkeiten notwendig sind, könnten diese als eigene Backend-Komponenten implementiert und über die API bereitgestellt werden. Für Formatkonvertierungen könnte der FME Server verwendet werden.	
DE22	Änderungen an den Daten sollen angezeigt werden können (Überarbeitungs modus)	o	Im Data-Explorer können Datensätze in Diagrammen überlagert, oder einzeln visualisiert werden.	
DI1	Es soll eine Nutzeroberfläche und ein Workflow gestaltet werden, um weitere Datenquellen anzubinden.		Unabhängig von Highcharts.	
DI2	Der Data-Input soll entweder durch die bestehenden Excel Templates		Unabhängig von Highcharts.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	der Redaktion oder falls möglich durch eine direkte Anbindung der Datenhaltung der datenhaltenden Stellen durchgeführt werden können.			
DI3	Daten sollen beim Import eine Qualitätssicherung durchlaufen		Unabhängig von Highcharts.	
DI4	Der Data-Input muss mit verschiedenen Eingangsformaten (Datenbanken, Excel, CSV, WebServices) umgehen können um Daten von verschiedenen datenhaltenden Stellen einzulesen.		Unabhängig von Highcharts.	
DI5	Metadaten, welche in den Datenquellen enthalten sind sollen mit ausgelesen werden.		Unabhängig von Highcharts.	
DI6	Während des Data-Inputs sollen zusätzliche Metadaten für einen Datensatz angereichert werden können.		Unabhängig von Highcharts.	
DI7	Während des Data-Inputs sollen Verlinkungen zu anderen Datensätzen im Data-Store möglich sein.		Unabhängig von Highcharts.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DI8	Mit dem Data-Input soll es möglich sein, mehrere Versionen von einem Datensatz einzulesen und im Data-Store zu speichern.		Unabhängig von Highcharts.	
DI9	Für Datensätze mit gleichbleibenden Strukturen soll ein automatisierter Importprozess möglich sein.		Unabhängig von Highcharts.	
DI10	Für Automatisierungen soll es möglich sein, zeitliche Ablaufpläne zu konfigurieren, um Datensätze regelmäßig zu importieren.		Unabhängig von Highcharts.	
RP1	Es muss ein redaktioneller Prozess zur Pflege der Daten zur Umwelt entwickelt werden, oder der DataCube muss in den bestehenden Prozess mit eingebettet werden.		Da Data-Explorer und Data-Output neu implementiert werden müssen, kann der redaktionelle Prozess von Anfang an mit in die Konzeption integriert werden. Grundsätzlich müssen die verschiedenen Freigabestufen für die Veröffentlichung berücksichtigt werden.	
RP2	Daten dürfen nur nach einer Überprüfung durch einen Data-Output auf der Webseite dargestellt werden.			
RP3	Data-Outputs müssen aufwandsarm in Artikel der UBA			

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Webseite eingebettet werden können.			
RP4	Data-Outputs dürfen nur nach vorheriger Freigabe veröffentlicht werden.			
RP5	Bei Anpassung der Daten eines Berichts/Artikels sollen Redakteure benachrichtigt werden.			
RP6	Der manuelle Aufwand des redaktionellen Prozesses soll soweit möglich reduziert werden.			
DS1	Es ist eine zentrale Datenhaltung mit allen Daten zur Umwelt aufzubauen.		Unabhängig von Highcharts.	Unabhängig von Highcharts.
DS2	Die Datenstrukturen müssen klar definiert werden. Es muss ein themenübergreifender Gesamtansatz für das Datenmodell ausgearbeitet werden, in den sich die verschiedenartigen Daten zur Umwelt einordnen.		Unabhängig von Highcharts.	
DS3	Das Datenmodell muss so konzipiert werden, dass es einerseits beliebig vertieft/		Unabhängig von Highcharts.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	detailliert und andererseits beliebig fachlich erweitert werden kann.			
DS4	Die datenhaltenden Stellen sollen ihre Daten möglichst automatisiert in den Data-Store einspielen können.		Unabhängig von Highcharts.	
DS5	Auch rückwirkend müssen Datenaktualisierungen und -erfassungen im Data-Store an die Fachsysteme (teil-) automatisiert übergeben werden können.		Unabhängig von Highcharts.	
DS6	Die Webseite soll Daten direkt aus dem Data-Store nutzen.		Unabhängig von Highcharts.	
DS7	Datenänderungen sollen automatisch an die Redakteure bzw. Fachexperten/-innen übermittelt werden.		Unabhängig von Highcharts.	
DS8	Die Daten müssen über Status so markiert sein, dass sofort erkennbar ist, in welchem Bearbeitungszustand sie sich gerade befinden.		Unabhängig von Highcharts.	
DS9	Dynamische Zugriffe auf die Daten müssen in Abhängigkeit des Status definiert		Unabhängig von Highcharts.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	und kontrolliert werden können.			
DS10	Es muss möglich sein, Daten aus verschiedenen Quellsystemen (insbesondere aus DESTATIS) zu übernehmen, diese einheitlich strukturiert zu verwalten, sodass sie dann als Dimensionen der unterschiedlichen Daten zur Umwelt im Data-Store genutzt werden können.		Unabhängig von Highcharts.	
DS11	Es muss möglich sein, für alle erfassten Datensätze Dimensionen zu definieren.		Unabhängig von Highcharts.	
DS12	Es muss möglich sein, die übernommenen Daten (konkrete Werte der Datensätze) den entsprechenden Dimensionen zuzuordnen.		Unabhängig von Highcharts.	
DS13	Die Werte müssen auch als Werte (Zahlen, Vektoren) gespeichert werden, sodass mit ihnen gerechnet werden kann.		Unabhängig von Highcharts.	
DS14	Es soll möglich sein, im Data-Store Berechnungen zu hinterlegen, die automatisch		Unabhängig von Highcharts.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	ausgeführt werden.			
DS15	Es soll möglich sein, bestimmte Berechnungen manuell anzustoßen.		Unabhängig von Highcharts.	
DS16	Es soll möglich sein, Berechnungen hierarchisch auszuführen. Das gilt auch für automatisierte Berechnungen.		Unabhängig von Highcharts.	
DS17	Es muss auch weiterhin möglich sein, extern mit den Daten zur Umwelt zu arbeiten und externe Daten mit ihren Bezügen zu den Ausgangsdaten im Data-Store einzuspielen.	-	Unabhängig von Highcharts. Es müsste jedoch ein Export/Import Workflow zur Bearbeitung von Daten außerhalb des Tools etabliert werden. Dies könnte z.B. durch ETL-Prozesse realisiert werden.	
DS18	Die Werte müssen historisch verwaltet werden.		Unabhängig von Highcharts.	
DS19	Bei hierarchischen Berechnungen müssen die Bezüge der Daten jederzeit wieder selektierbar sein.		Unabhängig von Highcharts.	
DS20	Objektklassen, die für die Dimensionierung herangezogen werden, müssen ebenfalls historisch verwaltet werden, um ältere Daten zur Umwelt auch		Unabhängig von Highcharts.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	weiterhin einordnen zu können.			

3.5.2.3 Bewertung

Durch den großen Anteil an eigenen Implementierungen ist der zu erwartende Arbeitsaufwand sehr hoch. Die Umsetzung aller Anforderungen ist sehr zeitintensiv und würde voraussichtlich über die initial geplante Projektphase hinaus verlaufen. Der Vorteil des Lösungsansatzes ist jedoch, dass alle Anwendungen für den Data Cube implementiert werden können und somit der vermutlich höchste Deckungsgrad zu den gegebenen Anforderungen erreicht werden könnte. Bei Bedarf können in diesem Ansatz Funktionalitäten iterativ ergänzt werden, wodurch zumindest einfache Diagramme voraussichtlich schnell im Projekt umgesetzt werden könnten.

Wie im vorherigen Kapitel 3.5.2.1 beschrieben, dient die Highcharts-Bibliothek hauptsächlich als Hilfsmittel zur Realisierung der Komponenten Data-Output und Data-Explorer. Im Gegensatz dazu wurde darauf hingewiesen, dass die Datenhaltung für den Data-Store und genauso die Data-Input Komponente unabhängig von der JavaScript Bibliothek implementiert werden müssten. Daher wäre eine Kombinationslösung mit einem weiteren Tool möglich, dass für die Datenhaltung und -verwaltung eingesetzt wird und somit die Anforderungen an die Data-Input und Data-Store Komponenten erfüllt. In diesem Kontext ist das Tool Mesap als Beispiel genannt, welches in Kapitel 3.5.3 näher beschrieben wird.

3.5.3 Mesap

Das Produkt Seven2one Plattform der Firma Seven2one Informationssysteme GmbH ist im UBA unter dem Namen Mesap bekannt und zum Beispiel im FG V 1.6 im Einsatz (vgl. Kapitel 2.3.2.3). In einer gemeinsamen Videokonferenz mit Mitarbeitenden des UBA (19.11.2021) wurden mögliche Einsatzszenarien von Mesap im Kontext des Data Cube diskutiert und die Abdeckung der Anforderungen eingeordnet. Der beschriebene Lösungsansatz in Kapitel 3.5.3.1 basiert maßgeblich auf diesem Gespräch und auf dem Anwender-Leitfaden des Herstellers. Im weiteren Verlauf wird konsequent der im UBA bekannte Name Mesap für die Seven2one Plattform verwendet.

3.5.3.1 Lösungsansatz

Mesap ist „die Basis aller individuellen Seven2one Lösungen“ (Heinz, 2020) und wird unter anderem im UBA als speziell angepasste Programmversion verwendet. In den betroffenen Abteilungen ist die Software für die Verarbeitung von Zeitreihen im Einsatz und bietet zudem Möglichkeiten einfacher statistischer Auswertungen. Auf Grundlage der Erfahrungen mit Mesap im UBA ist festzuhalten, dass diese Lösung hauptsächlich für die Komponenten Data-Store und Data-Input des Data Cube Projektes von Relevanz ist. Während vereinzelte Anforderungen für die Datenexploration im Kontext des Data-Explorers abgedeckt werden könnten, stehen kaum Funktionen bereit, die für den web-basierten Data-Output verwendbar wären.

Mesap ist als eine sogenannte Middleware zu sehen, mit der unter anderem der Datenimport, die Datensuche und der Datenexport realisiert werden können. Dem Anwendenden steht hierzu eine graphische Benutzeroberfläche zur Verfügung. Für technisch-versierte Anwender ist es auch möglich, die zugrunde liegenden Datenbanken über Skripte oder beispielsweise SQL-

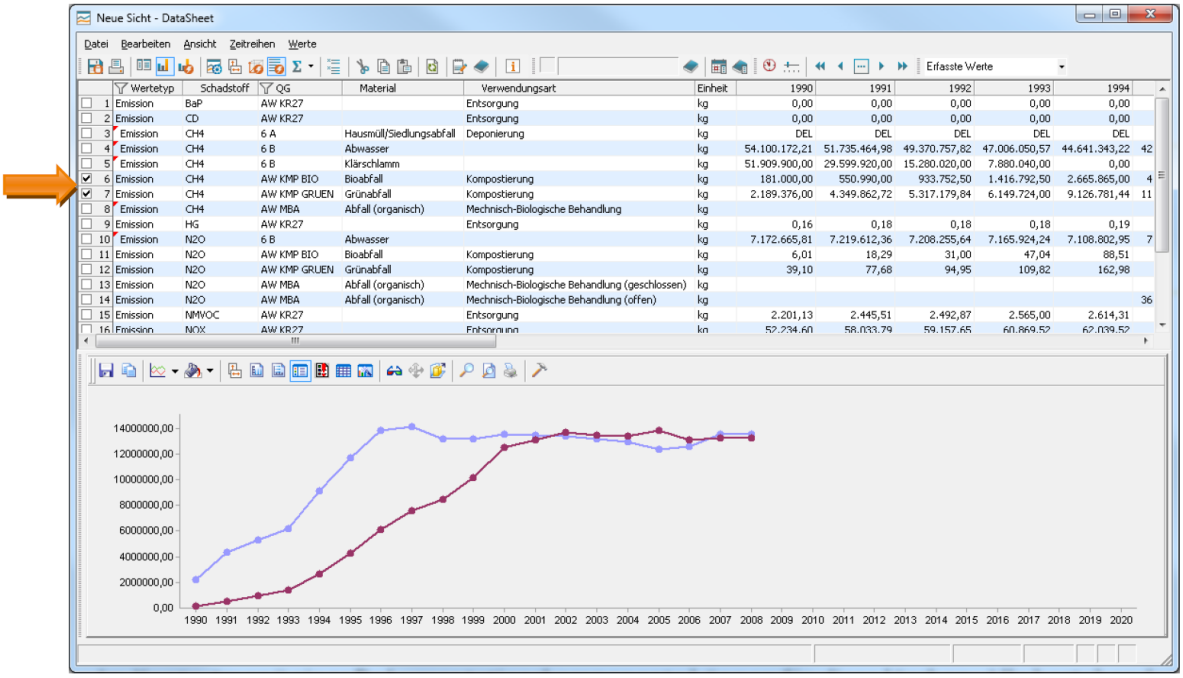
Statements anzusprechen. Insgesamt ist somit mit Mesap die Orchestrierung von Daten abzubilden, während die eigentliche Datenhaltung zum Beispiel in Microsoft SQL Server oder Oracle Datenbanken erfolgt. Für den Data Cube ist sowohl die Datenhaltung innerhalb einer Datenbank als auch die Verwendung von mehreren Datenbanken für den Data-Store möglich. In den bestehenden Anwendungsszenarien sind die vollständigen Zeitreihen und Tabellen auf der flachsten Hierarchiestufe in der Datenbank enthalten. Während in den derzeitigen Anwendungen Einträge häufig manuell eingepflegt werden, besteht auch die Möglichkeit Schnittstellen (z.B. .Net API) oder Webservices für den Import zu nutzen. Darüber hinaus wurde darauf hingewiesen, dass der Import von wohlformatierten CSV-Dateien ebenso abgedeckt wird. Für komplexere Datenstrukturen muss eine Datenaufbereitung durch eine ETL-Software durchgeführt werden. Für einen direkten Import durch ETL Prozesse müssten die verfügbaren APIs genauer analysiert werden.

Im Kontext Datenexploration ist die Auflistung und Suche von Daten und Zeitreihen sowie deren Zusammenhänge in Form von Bäumen mit Mesap abbildbar.

Die Baumstrukturen werden genutzt, um verschiedene Hierarchien abbilden zu können.

In Mesap werden multidimensionale Daten mittels Dimensionen beschrieben. Für diese Dimensionen werden dann sogenannte Deskriptoren hinterlegt. Dies sind die konkreten Ausprägungen der Dimensionen. Die Deskriptoren und Dimensionen können in Hierarchiebäumen strukturiert werden (vgl. Kapitel 7.1 in (Seven2one, o. J.)). Durch diese hierarchischen Strukturen wiederum werden Aggregationen und Deaggregationen der Daten unterstützt. Obwohl in diesem Kontext nicht explizit von Cubes gesprochen wird, können mit Mesap ähnliche Funktionalitäten durch die flexible Darstellung in Bäumen abgebildet werden. Ebenso sind Operationen, vergleichbar zu Cube-Funktionalitäten wie slice oder drill-down, implementiert. Mittels sogenannter DataSheets ist es möglich, Wertefilter abzubilden, um benutzerspezifische Sichten auf die Daten zu realisieren. Zusätzlich zu Aggregationsberechnungen ist es weiterhin möglich, einfache Berechnungsvorschriften zu hinterlegen. Nach derzeitigem Stand werden die berechneten Werte nicht persistiert, sondern ad-hoc berechnet. Hiermit sind auch Daten-Interpolation oder -Extrapolationen möglich. Hierzu stehen auch zusätzliche Add-Ins wie der Analyst, der Calculator oder Converter bereit. Für tiefergehende statistische Berechnungen existiert ein Zusatzmodul für Berechnungen mittels Matlab oder R. Dieses Modul müsste ggf. nachlizensiert werden.

Abbildung 7: Grafik Panel in Mesap

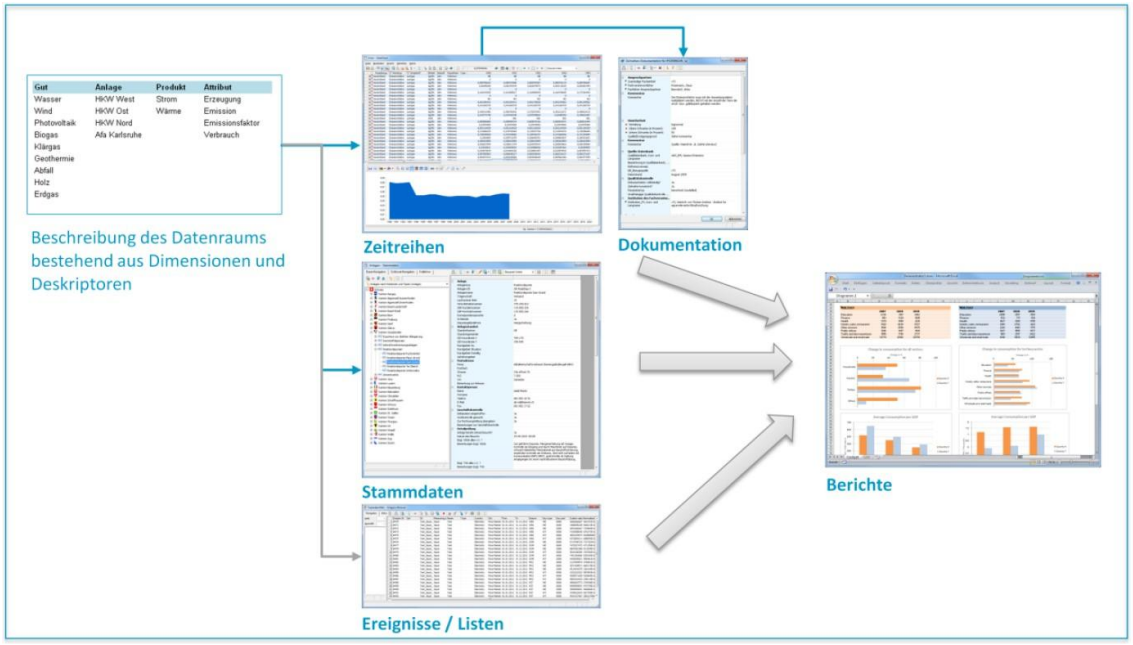


Quelle: Kapitel 2.3.8 in (Seven2one, o. J.)

Insgesamt kann damit ein gewisser Teil der Anforderungen für den Data-Explorer abgedeckt werden. Mesap wird als lokale Anwendung betrieben und bietet somit Datenexploration für interne Mitarbeitende. Durch eine gute Anbindung an Excel können Daten einfach bearbeitet und visualisiert werden. Während es zwar möglich ist, Daten in Form von z. B. CSV-Dateien zu exportieren und auch einfache Diagramme zur Visualisierung bereitzustellen (vgl. Grafik Panel in Abbildung 7), ist der Export von Visualisierungen nicht abgedeckt. Folglich ist es ebenso nicht möglich, Dashboards aus mehreren Data-Outputs zu generieren. Das bedeutet, dass interaktive Diagramme zur Datenerkundung nicht mit Mesap für externe Nutzende bereitgestellt werden können. An dieser Stelle ist nicht abschließend zu klären, ob die vorhandenen APIs von Mesap direkt für eine Eigenentwicklung mittels Highcharts (s. Kapitel 3.5.2) implementiert werden könnten. Bei Bedarf müsste eine eigene Backend-Komponente implementiert werden, die Benutzerauthentifizierungen berücksichtigt und daraufhin die Datenabfrage ermöglicht.

Mesap bietet weiterhin Funktionalitäten zur Versionierung und Historisierung. Während Strukturänderungen in einem Datenbankprotokoll festgehalten werden, wird für Datensätze eine Änderungshistorie bereitgestellt. Diese sind in den Metadaten, die bei Mesap als Dokumentationen bekannt sind, enthalten. Da für diese Datenobjekte eine benutzerdefinierte Struktur verwendet werden kann (Seven2one, o. J.), ist es hiermit möglich, Felder wie Herkunft der Daten, Ansprechpartner oder weitere Kennziffern zu pflegen. Für Versionsabfragen können erneut die DataSheets verwendet werden, um Wertefilter zum Beispiel auf Änderungsdaten zu setzen.

Abbildung 8: Grundkomponenten in Mesap



Quelle: seven2one Informationssysteme GmbH: “Mesap - Leitfaden für den Anwender”, Kap.7

Abbildung 8 gibt einen Überblick über die in Mesap verwalteten Objekte. Der sogenannte Datenraum besteht aus Dimensionen und Deskriptoren (die Dimensionswerte). Deskriptoren werden Dimensionen zugeordnet und in Hierarchiebäumen organisiert. Die Datenverwaltung ist generisch, in Mesap-internen Strukturen. Durch die Verlinkung von Deskriptoren zu Bäumen können beliebig tiefe Hierarchien gebildet werden. Über die Hierarchien können Berechnungsvorschriften verwaltet werden. Die eigentlichen Werte werden als Zeitreihen verwaltet, in regelmäßiger Zeitauflösung (Periode). Eine Zeitreihe verweist über den „Multi-dimensionalen Schlüssel“ auf die einzelnen Deskriptoren der einzelnen Dimensionen. Darüber hinaus erhält die Zeitreihe „Ausprägungen“ (Zeitauflösung, Einheit, Szenarien = „Hypothese“).

Abbildung 9: Zeitreihen in Mesap

ID & Name		Multi-dimensionaler Schlüssel			Ausprägung			Werte		
ZR-ID	ZR-Name	Deskriptor	..	Deskriptor	Zeitauflösung	Einheit	Hypothese	Zeitpunkt	Zeitpunkt	Zeitpunkt
								Wert	Wert	Wert
#29		CO2		Emission	Jahr	t	Ref	87.000	87.531	100.395

Quelle: seven2one Informationssysteme GmbH: “Mesap - Leitfaden für den Anwender”, Kap.7

Abbildung 9 stellt schematisch einen Zeitreihen-Datensatz dar. Dokumente sind Datenobjekte, die den Zeitreihen und einzelnen Zeitreihenwerten zugeordnet werden können. Dokumente können in benutzerdefinierten Strukturen abgelegt werden. Mesap kann die Stammdaten in beliebigen Formaten ablegen. Das sind vor allem Listen zu Parametern, Dimensionen, sowie Datenbestände mit einfacher Datenstruktur. Aus den Zeitreihen, Stammdaten, Ereignisse/Listen können in Mesap Berichte erstellt werden.

3.5.3.2 Anforderungstabelle

In der folgenden Tabelle werden alle Anforderungen an den Data Cube durch die Lösungskomponente beschrieben und nach der Klassifikation aus Kapitel 3.5 bewertet.

Tabelle 4: Anforderungstabelle der Lösungskomponente Mesap

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
AA1	Die Software soll erweiterbar/ anpassbar sein, falls nicht alle Anforderungen abgedeckt werden können (Anpassung von Design, aber auch Funktionalitäten)	o	Mesap ist eine Software der seven2one GmbH, die an den Anforderungen der Nutzenden vom Hersteller angepasst wird. Darüber hinaus ist eine API für eigene Entwicklungen verfügbar.	https://www.seven2one.de/news/techstack-go-neue-plattform-generation-einsatzbereit/
AA2	Die Anforderungen müssen durch die Software effizient implementiert werden können.	o	Standard-Software kann direkt verwendet werden. Die Integration für andere Komponenten (z.B. Data-Output) muss jedoch implementiert werden.	
AA3	Die Software soll mit geringem Aufwand in Betrieb genommen werden.	+	Standard-Software, ist bereits im UBA im Einsatz	
AA4	Die Software muss für Anwender und Administratoren ausreichend dokumentiert sein.	+	Leitfaden für den Anwender und weitere Dokumentationen liegen vor.	
AA5	Die Software ist bereits ausgereift und kann direkt verwendet werden.	-	Die Software wird bereits vom UBA eingesetzt und seit einigen Jahren verwendet. Mesap wird in wenigen Jahren durch ein neues Programm des Herstellers abgelöst.	
AA6	Es gibt eine aktive Community mit Diskussionsforen und Beispielen zur Anwendung.	-	Die Lösungen sind angepasst. Für Fragen steht der Hersteller zur Verfügung. Eine Community existiert nicht. Es gibt jedoch viel Erfahrungen und Beispiele durch den Einsatz im UBA.	
AA7	Die Software ist kostenfrei nutzbar, oder die Kosten sind in einem für das Projekt	+	Die Software ist im UBA im Einsatz. Das FG kann über Kosten/Wirtschaftlichkeit Auskunft geben.	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	vertretbaren Rahmen.			
AA8	Die Software kann im UBA Rechenzentrum betrieben werden	+	Wird bereits im UBA auf dem MS SQL Cluster betrieben. Gegebenenfalls ist eine weitere Instanz notwendig.	
AA9	Die Software kann ohne Weiteres gratis getestet werden	o	Die Software ist im UBA im Einsatz.	
DO1	Es muss eine Visualisierung von Daten möglich sein, bei denen eine feste Ansicht auf die Daten und eine feste Darstellungsform durch die Redaktion vorgegeben wird.		Unabhängig von Mesap. Die API von Mesap müsste für eine Server-Komponente evaluiert werden, um eine API für Data-Outputs zur Verfügung zu stellen.	-
DO2	Daten müssen als Tabelle dargestellt werden können.		Unabhängig von Mesap	
DO3	Daten innerhalb der Tabellendarstellung sollen gefiltert werden können.		Unabhängig von Mesap	
DO4	Daten innerhalb der Tabellendarstellung sollen sortiert werden können.		Unabhängig von Mesap	
DO5	Aggregationen von Datensätzen sollen vorgenommen oder vorberechnete Aggregationen angezeigt werden können.		Darstellung ist unabhängig von Mesap. Deskriptoren können hierarchisch in Bäumen verwaltet werden. Auf diese Weise sind Aggregationen möglich. Extrapolationen/ Interpolationen sind umgesetzt. Diese müssten über die API abrufbar sein.	
DO6	Deaggregationen von Datensätzen sollen vorgenommen		Darstellung ist unabhängig von Mesap. Deskriptoren können hierarchisch in Bäumen verwaltet werden. Auf diese	"Leitfaden", Kap.3.4

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	oder vorberechnete Deaggregationen angezeigt werden können.		Weise sind Aggregationen möglich. Extrapolationen/ Interpolationen sind umgesetzt. Diese müssten über die API abrufbar sein.	
DO7	Zeitreihen sollen eingeschränkt werden können (Filterung durch min. und max. Datum).		Unabhängig von Mesap	
DO8	Einheiten sollen dynamisch umgerechnet werden, oder durch vordefinierte Daten in anderen Einheiten angezeigt werden können.		Darstellung ist unabhängig von Mesap. Umrechnungen können hinterlegt werden und müssten über die API abrufbar sein.	
DO9	Die Darstellungsform der Abbildungen soll durch den Nutzenden frei gewählt werden können. Gewünschte Darstellungsformen sind: Liniendiagramm, Balkendiagramm, Tortendiagramm, Baumdiagramme, Abweichungen, Korrelationen und Streudiagramme, Häufigkeitsverteilungen, Nominale Vergleiche (wie Blasendiagramme oder Heatmaps), Fluss und Sankey-Diagramme, Netzwerkdiagramme)		Unabhängig von Mesap	
DO10	Die Visualisierungen in Diagrammen/Abb		Unabhängig von Mesap. Es müsste ein eigener Data-Output implementiert werden, oder eine bestehende	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	ildungen (Data-Outputs) müssen direkt in Artikel auf der UBA-Webseite (Drupal CMS) eingebunden werden können		Lösungskomponente müsste entsprechend angebunden werden. Je nach Implementierung müsste eine Integration in Drupal durchgeführt werden.	
DO11	Die Visualisierungen in Diagrammen/Abbildungen (Data-Outputs) soll in der zukünftigen umwelt.info Webseite möglich sein.		Unabhängig von Mesap	
DO12	Die Implementierung soll möglichst CMS-offen (unabhängig von dem bestehenden Drupal System) erfolgen. Es soll geprüft werden ob Inhalte z.B. als reponsive iFrames oder vergleichbares eingebettet werden können.		Unabhängig von Mesap	
DO13	Funktionalitäten die nicht CMS-offen implementiert werden, sind als Drupal-Modul (unter Beachtung der Drupal Code-Konventionen) zu entwickeln in vollständig lauffähig in das CMS der UBA Webseite zu integrieren.	+	Drupal Standards können bei einer Drupal-Modul-Implementierung berücksichtigt werden.	
DO14	Es muss möglich sein, Dashboards im Drupal CMS		Unabhängig von Mesap	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	einzubinden, die aus mehreren Data-Outputs bestehen			
DO15	Dashboards und Data-Outputs sollen auf verschiedenen Display Größen (Mobilgeräte und größere Bildschirme) gut angezeigt werden.		Unabhängig von Mesap	
DO16	Daten und Abbildungen sollen in unterschiedlichen Formaten zum Download angeboten werden. Daten: CSV, Excel Abbildungen: PNG/JPEG	o	Es ist kein Data-Output für das Web durch Mesap verfügbar. Ggf. könnte für eine zu implementierende Download Funktion die Mesap Web Services API verwendet werden.	
DO17	Datensätze die zu einem Thema gehören sollen erkundet werden können. Das heißt, dass die Daten sowie verlinkte Datensätze angezeigt werden.	o	Es ist kein Data-Ouput für Mesap verfügbar. Die Datenhaltung sieht jedoch auch keine Verknüpfungen der Zeitreihen untereinander vor. Dimensionen können in mehreren Zeitreihen verwendet werden, Hierarchien sind über die Baumstruktur anlegbar. In einem zu implementierenden Data-Output könnten daher voraussichtlich Bäume erkundbar gemacht werden.	
DO18	Verknüpfte Datensätze sollen in der Visualisierung mit hinzugefügt werden können, um Daten vergleichen zu können.		Unabhängig von Mesap	
DO19	Das Corporate Design ist bei der Gestaltung aller		Unabhängig von Mesap	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	Nutzeroberflächen zu beachten.			
DO20	Es ist zu berücksichtigen wie Data-Outputs auch in die Datensuche des UBA eingebunden werden können.	o	Wenn Data-Outputs als Drupal-Module implementiert werden, kann der Standard Drupal-Mechanismus verwendet werden. Hierzu müssen beim Erstellen der Inhalte in Drupal entsprechende Keywords vergeben werden.	
DE1	Über den Data-Explorer müssen Datensätze des Data Stores auffindbar gemacht werden.	+	Suchmöglichkeiten vorhanden	"Leitfaden", Kap.2
DE2	Metadaten von Datensätzen sollen angezeigt werden können.	+	Es können verschiedene Metadaten (Forschungskennziffer, Ansprechpartner, Unsicherheit) gespeichert und angezeigt werden. Dazu Anzeige von Dimensionen und Deskriptoren.	
DE3	Datensätze sollen als interaktive Diagramme angezeigt werden können.	+	Es werden einfache interaktive Diagramme angeboten.	"Leitfaden", Kap.2.3.8
DE4	Datensätze sollen tabellarisch angezeigt werden können.	+	Tabellen sind die Standardansicht für Zeitreihen.	"Leitfaden", Kap.2
DE5	Eine Definition von zusammengehörigen Datensätzen für einen Cube soll möglich sein.	+	In Mesap werden Daten entlang von Dimensionen und Deskriptoren organisiert. Dadurch entsteht eine hohe Flexibilität in der Zusammenstellung von Datensätzen.	"Leitfaden", Kap.2
DE6	Es soll möglich sein, alle Cubes aufzulisten.	+	Die Datensätze können selektiert und aufgelistet werden.	"Leitfaden", Kap.2
DE7	Es soll möglich sein alle Datensätze aufzulisten, die in einem Cube	+	Die Datensätze können selektiert und aufgelistet werden.	"Leitfaden", Kap.2

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	verwendet werden.			
DE8	Es soll möglich sein für einen Datensatz anzuzeigen, in welchem Cube er verwendet wird.	-	Nicht möglich.	
DE9	Für jeden Datensatz sollen fachlich begründete Dimensionen ausgewählt werden können.	+	Die Zuordnung von Dimensionen zu Bäumen ist vorgesehen.	
DE10	Hierarchische Datensätze sollen miteinander verknüpft werden können.	+	Hierarchien können über die Baumstrukturen abgebildet werden.	
DE11	Datenauszügen für einzelne Datensätze sollen definiert werden können.	+	Datenauszüge können über Filter- und Selektionsfunktionen erstellt werden.	"Leitfaden", Kap.2
DE12	Eine Konfiguration von Datensätzen / Dimensionen für einzelne Data-Outputs soll ermöglicht werden. Es soll konfigurierbar sein, welche weiteren Datensätze zu einem Data-Output potenziell hinzugeladen werden dürfen.		Unabhängig von Mesap. Es ist keine Data-Output Funktion verfügbar.	
DE13	Konfiguration der initialen und konfigurierbaren Darstellungsform		Unabhängig von Mesap. Es ist keine Data-Output Funktion verfügbar.	
DE14	Konfiguration weiterer Optionen des Data-Outputs für		Unabhängig von Mesap.	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	die Nutzenden: - Daten hinzuladen - Einheiten umrechnen / Datensatz mit anderen Einheiten laden - Aggregationen berechnen / Datensätze mit anderen Aggregationen laden - Erlaubte Downloads			
DE15	Für jeden Data-Output sollen die Zusatz-Funktionalitäten (z.B. Aggregationen berechnen, Daten hinzuladen, …) bei Bedarf deaktiviert werden können.	o	(De-)Aggregationen können pro Darstellung konfiguriert werden. Hinzuladen von Daten oder Umrechnen von Einheiten ist nicht konfigurierbar.	
DE16	Konfiguration von Dashboards (Kombination aus mehreren Data-Outputs)		Unabhängig von Mesap.	
DE17	Der Data-Explorer soll außerhalb des UBA CMS implementiert werden können und hat kein spezifisches CMS zur Voraussetzung.	+	Es handelt sich bei Mesap um eine Desktop Anwendung und wird außerhalb des CMS betrieben.	
DE18	Für externe Analysetools soll der Zugriff auf die Daten über eine API ermöglicht werden.	o	Mesap bietet ein Web Services Modul sowie Programmierschnittstellen, auf welchen APIs bereitgestellt werden könnten. Die Funktionen des Web Services Modul sind nicht bekannt. Gegebenenfalls ist eine eigene Entwicklung notwendig.	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
DE19	Für externe Analysetools soll der Zugriff auf die Daten über einen Export ermöglicht werden.	+	Exporte sind als Excel oder XML möglich. Gegebenenfalls sind über weitere Zusatzmodule noch weitere Formate möglich. Alternativ könnte die API zum Export verwendet werden.	
DE20	Die Konfiguration der Data-Outputs und Data Cubes soll über ein Webinterface möglich sein.	-	Nicht möglich. Mesap als Data-Explorer ist eine Desktop-Anwendung. Mesap hat keine Data-Output Komponente.	
DE21	Es soll möglich sein Zusatzdateien für Berichte (Bilder in verschiedenen Formaten, Daten) herunterzuladen.	o	Daten können exportiert werden. Diagramme oder ähnliches sollten über eine Data-Output Komponente erstellt werden und sind daher unabhängig von Mesap. Zusätzlich können Über die Dokumentation zu Zeitreihen und Zeitreihenwerten nutzerspezifisch strukturierte Dokumente angebunden werden.	"Leitfaden", Kap.7
DE22	Änderungen an den Daten sollen angezeigt werden können (Überarbeitungs modus)	+	Protokollierung der Änderungen der Werte	
DI1	Es soll eine Nutzeroberfläche und ein Workflow gestaltet werden um weitere Datenquellen anzubinden.	+	Ist als Funktionalität vorhanden.	"Leitfaden", Kap.2
DI2	Der Data-Input soll entweder durch die bestehenden Excel Templates der Redaktion oder falls möglich durch eine direkte Anbindung der Datenhaltung der datenhaltenden Stellen	+	Daten können aus verschiedenen Quellen (Excel, CSV, WebServices) übernommen werden. Gegebenenfalls sind ETL Prozesse zu verwenden.	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	durchgeführt werden können.			
DI3	Daten sollen beim Import eine Qualitätssicherung durchlaufen	o	Grundsätzlich vorhanden, beim UBA jedoch kaum in Verwendung. Gegebenenfalls muss eine weitere ETL Software verwendet werden.	
DI4	Der Data-Input muss mit verschiedenen Eingangsformaten (Datenbanken, Excel, CSV, WebServices) umgehen können um Daten von verschiedenen datenhaltenden Stellen einzulesen.	+	Daten können aus verschiedenen Quellen (Excel, CSV, WebServices) übernommen werden.	
DI5	Metadaten, welche in den Datenquellen enthalten sind sollen mit ausgelesen werden.	o	Metadaten können über die GUI eingetragen werden. Es gibt keinen automatischen Import.	
DI6	Während des Data-Inputs sollen zusätzliche Metadaten für einen Datensatz angereichert werden können.	o	Metadaten können über die GUI eingetragen werden. Es gibt keinen automatischen Import.	
DI7	Während des Data-Inputs sollen Verlinkungen zu anderen Datensätzen im Data-Store möglich sein.	o	Dimensionen können mehrfach verwendet werden; Baumstrukturen sind möglich.	
DI8	Mit dem Data-Input soll es möglich sein, mehrere Versionen von einem Datensatz einzulesen und im	+	Versionierung ist möglich	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	Data-Store zu speichern.			
DI9	Für Datensätze mit gleichbleibenden Strukturen soll ein automatisierter Importprozess möglich sein.	o	Es ist keine Automatisierung möglich. Über die APIs könnte jedoch mit einem ETL Tool voraussichtlich eine Automatisierung implementiert werden.	
DI10	Für Automatisierungen soll es möglich sein, zeitliche Ablaufpläne zu konfigurieren, um Datensätze regelmäßig zu importieren.	o	Es ist keine Automatisierung möglich. Über die APIs könnte jedoch mit einem ETL Tool voraussichtlich eine Automatisierung implementiert werden.	
RP1	Es muss ein redaktioneller Prozess zur Pflege der Daten zur Umwelt entwickelt werden, oder der DataCube muss in den bestehenden Prozess mit eingebettet werden.	o	Der redaktionelle Prozess ist unabhängig von Mesap.	
RP2	Daten dürfen nur nach einer Überprüfung durch einen Data-Output auf der Webseite dargestellt werden.		Unabhängig von Mesap	
RP3	Data-Outputs müssen aufwandsarm in Artikel der UBA Webseite eingebettet werden können.		Unabhängig von Mesap	
RP4	Data-Outputs dürfen nur nach vorheriger		Unabhängig von Mesap	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	Freigabe veröffentlicht werden.			
RP5	Bei Anpassung der Daten eines Berichts/Artikels sollen Redakteure benachrichtigt werden.	-	Nicht möglich. Änderungen in den Zeitreihen können aber protokolliert werden.	
RP6	Der manuelle Aufwand des redaktionellen Prozesses soll soweit möglich reduziert werden.		Unabhängig von Mesap.	
DS1	Es ist eine zentrale Datenhaltung mit allen Daten zur Umwelt aufzubauen.	+	Mesap ist eine "Middleware", die die Daten orchestriert. Die Daten werden als Zeitreihen abgelegt.	"Leitfaden", Kap.7
DS2	Die Datenstrukturen müssen klar definiert werden. Es muss ein themenübergreifender Gesamtansatz für das Datenmodell ausgearbeitet werden, in den sich die verschiedenartigen Daten zur Umwelt einordnen.	+	Alle Daten können als Bäume abgelegt werden. Dimensionen die in mehreren Datensätzen gleich sind, können wiederverwendet werden, wodurch eine Kombination verschiedener Datensätze möglich ist.	"Leitfaden", Kap.7
DS3	Das Datenmodell muss so konzipiert werden, dass es einerseits beliebig vertieft/ detailliert und andererseits beliebig fachlich erweitert werden kann.	+	Erweiterungen/Vertiefungen sind möglich, Detaillierungen auch über Baumstrukturen.	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
DS4	Die datenhaltenden Stellen sollen ihre Daten möglichst automatisiert in den Data-Store einspielen können.		Unabhängig von Mesap. Automatisierungen müssen durch ein ETL Tool durchgeführt werden.	
DS5	Auch rückwirkend müssen Datenaktualisierungen und -erfassungen im Data-Store an die Fachsysteme (teil-) automatisiert übergeben werden können.	+	Export der Daten ist möglich.	-
DS6	Die Webseite soll Daten direkt aus dem Data-Store nutzen.		Unabhängig von Mesap. Für Data-Output muss eine API implementiert werden. Durch diese muss ein Zugriff auf die Daten möglich sein.	-
DS7	Datenänderungen sollen automatisch an die Redakteure bzw. Fachexperten/-innen übermittelt werden.	-	Nicht möglich	
DS8	Die Daten müssen über Status so markiert sein, dass sofort erkennbar ist, in welchem Bearbeitungszustand sie sich gerade befinden.	o	Auf Werteebene gibt es mehrere Metadaten die ggf. für die Versionsabfrage verwendet werden könnten.	
DS9	Dynamische Zugriffe auf die Daten müssen in Abhängigkeit des Status definiert und kontrolliert werden können.	o	s. DS8	
DS10	Es muss möglich sein, Daten aus	+	Dimensionen, die in mehreren Datensätzen gleich sind,	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	verschiedenen Quellsystemen (insbesondere aus DESTATIS) zu übernehmen, diese einheitlich strukturiert zu verwalten, sodass sie dann als Dimensionen der unterschiedlichen Daten zur Umwelt im Data-Store genutzt werden können.		können wiederverwendet werden, wodurch eine Kombination verschiedener Datensätze möglich ist.	
DS11	Es muss möglich sein, für alle erfassten Datensätze Dimensionen zu definieren.	+	Zeitreihen bestehen aus n Dimensionen.	"Leitfaden", Kap.7
DS12	Es muss möglich sein, die übernommenen Daten (konkrete Werte der Datensätze) den entsprechenden Dimensionen zuzuordnen.	+	Ist möglich.	"Leitfaden", Kap.7
DS13	Die Werte müssen auch als Werte (Zahlen, Vektoren) gespeichert werden, sodass mit ihnen gerechnet werden kann.	+	Die Speicherung geschieht ausschließlich als Werte.	
DS14	Es soll möglich sein, im Data-Store Berechnungen zu hinterlegen, die automatisch ausgeführt werden.	+	Grundrechenarten können direkt am Wert definiert werden; Nutzung des Calculators (mit Statistik-Bibliothek); ansonsten auch externe Berechnungen möglich.	
DS15	Es soll möglich sein, bestimmte Berechnungen	+	Berechnungen können automatisch erfolgen, aber	

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
	manuell anzustoßen.		auch manuell angestoßen werden.	
DS16	Es soll möglich sein, Berechnungen hierarchisch auszuführen. Das gilt auch für automatisierte Berechnungen.	+	Über die Baumstruktur umgesetzt.	
DS17	Es muss auch weiterhin möglich sein, extern mit den Daten zur Umwelt zu arbeiten und externe Daten mit ihren Bezügen zu den Ausgangsdaten im Data-Store einzuspielen.	+	Die Daten können jederzeit extern verarbeitet werden und dann wieder in der Zeitreihe zur Verfügung stehen.	-
DS18	Die Werte müssen historisch verwaltet werden.	+	Änderungen der Werte werden protokolliert.	
DS19	Bei hierarchischen Berechnungen müssen die Bezüge der Daten jederzeit wieder selektierbar sein.	o	über die Protokollierung	
DS20	Objektklassen, die für die Dimensionierung herangezogen werden, müssen ebenfalls historisch verwaltet werden, um ältere Daten zur Umwelt auch weiterhin einordnen zu können.	-	Nicht möglich.	

3.5.3.3 Bewertung

Mesap wird im UBA schon seit einigen Jahren eingesetzt und von den Nutzenden sehr positiv bewertet. Damit könnte beim Einsatz von Mesap für den Data Cube auf reichhaltige technische und fachliche Erfahrungen zurückgegriffen werden. Ebenso wurde darauf hingewiesen, dass das Tool besonders kosteneffizient betrieben werden kann.

Der Einsatz von Mesap ist insbesondere im Kontext des Data-Stores und des Data-Explorers relevant. Für detaillierte Informationen zu den Integrationsmöglichkeiten von Mesap für diese Teilkomponenten des Data Cube wird auf Kapitel 3.5.3.1 verwiesen. In diesem Kapitel zur Beschreibung des Lösungsansatzes auf Basis von Mesap wurde bereits herausgestellt, dass Mesap keine besondere Data-Output Komponente implementiert, die für interaktive Diagramme und Dashboards für externe Nutzende verwendet werden könnte. Damit kann mittels Mesap ohne weiteren Entwicklungsaufwand oder eine andere Zusatzlösung keine Datenerkundung auf der UBA Webseite bereitgestellt werden. In diesem Kontext ist eine Erweiterung auf Grundlage einer Diagrammbibliothek wie zum Beispiel Highcharts möglich, um interaktive Data-Outputs zu generieren. Hierzu müssten die Schnittstellen von Mesap verwendet werden, um Daten für Visualisierungen in einer Fremdkomponente bereitzustellen. Da Mesap generell hierarchische Strukturen und Berechnungen unterstützt, sollte dies in einer API für eine (grafische) Weiterverarbeitung der Daten ebenso abgebildet werden können.

Eine Kombinationslösung aus Mesap und .Stat Suite ist vermutlich nicht sinnvoll, da Mesap keine SDMX-Schnittstellen für Import- oder Exportprozesse bereitstellt. Diese wären jedoch für den Einsatz der .Stat Suite zwingend erforderlich. Um das passende SDMX-Format bereitzustellen, wäre ein ETL-Prozess oder eine eigene API zur Transformation denkbar. Bei Möglichkeiten bedeuten jedoch zusätzliche Eigenentwicklungen.

Abschließend ist darauf hinzuweisen, dass die Weiterentwicklung der Mesap Technologie durch den Hersteller ausläuft und daher Wartungs- und Update-Prozesse seitens des Herstellers nur noch zeitlich begrenzt verfügbar sind. Es ist als ein besonderes Risiko zu bewerten, eine Software einzusetzen, die zeitnah seitens des Herstellers nicht mehr unterstützt wird. Hierdurch könnten nicht zuletzt Sicherheitslücken entstehen und den sicheren Betrieb des Data Cube gefährden. Die Nachfolgegeneration von Mesap wird bereits durch den Hersteller vertrieben. Der sogenannte Seven2one TechStack behält laut Hersteller Webseite die „Stärken des hochflexiblen Datenmodells und baut diese noch aus“ (Heinz, 2020). Darüber hinaus wird damit eine reine webbasierte Lösung implementiert, die sowohl cloudfähig ist als auch container-basiert installiert werden kann. Dazu soll eine Graph QL API in der neuen Version zum Einsatz kommen. Der Einsatz als on-premise Lösung wird aber weiterhin gegeben sein. Während davon auszugehen ist, dass Grundfunktionalitäten von Mesap in der neuen Software weiterhin Bestand haben werden, ist es nicht abschätzbar, ob eine höhere Anforderungsabdeckung bezüglich des Data-Explorers und insbesondere auch der Data-Outputs erzielt werden könnte. Hierfür wäre eine weitere Anforderungsanalyse für den TechStack notwendig. Zudem sollte dann in dem Kontext eruiert werden, ob weitere Fachgebiete auf die neue Technologie setzen würden, um den Vorteil von Synergieeffekten weiterhin nutzen zu können. Während der Einsatz von Mesap als besonders kosteneffizient beschrieben wurde, sind anfallende Kosten für die Nutzung des TechStack derzeit nicht abschätzbar.

3.5.4 Sisense

3.5.4.1 Lösungsansatz

Sisense ist eine kommerzielle Software zur Datenanalyse und Visualisierung. Die Anwendung hat einen starken Fokus auf „Embedded Analytics“ und ist bewusst zur Integration in andere

Lösungen entwickelt. Im Folgenden wird die Verwendung von Sisense als mögliche Lösungskomponente für das Data Cube Projekt erläutert.

Sisense kann Daten aus verschiedensten Datenquellen (Datenbanken oder auch dateibasierte Daten) (Sisense, o. J. - h) als Live-Datensätze anbinden. Dadurch kann die Data-Store Komponente des Data Cube als unabhängig von Sisense betrachtet werden. Grundsätzlich können beliebige Datenmodelle verwendet werden, die dann durch Sisense angebunden werden. Data-Input und Data-Store können somit mit anderen Tools realisiert werden, die jeweils die entsprechenden Anforderungen abdecken. Eine große Stärke von Sisense ist zudem die Definition von sogenannten ElastiCubes (Sisense, o. J. - d). ElastiCubes beschreiben eine proprietäre Datenbank von Sisense die für analytische Prozesse und häufige Abfragen durch BI-Anwendungen optimiert sein soll. In ElastiCubes können Datensätze aus verschiedenen Quellen über eine grafische Benutzeroberfläche zu Cubes zusammengefasst werden. Der Hersteller beschreibt die Technologie selbst als eine flexiblere Alternative zu OLAP-Cubes, da keine pre-aggregationen notwendig sind und trotzdem verschiedene Datensätze miteinander verschnitten werden können (Sisense, o. J. - d).

Das Zusammenfassen von Datensätzen als ElastiCubes ist bereits eine Anforderung an die Data-Explorer Komponente. Hierfür ist zusätzlich eine Suche verfügbar, um alle Datensätze im Data-Store aufzufinden (Sisense, o. J. - f). Nach der Definition der Datenquellen können Dashboards erzeugt werden. Sisense unterscheidet nicht zwischen einzelnen Data-Outputs und Dashboards. Es können jedoch beliebig viele Widgets pro Dashboard definiert werden. Ein Dashboard mit nur einem Widget (Sisense, o. J. - b) könnte demnach auch als einfacher Data-Output verwendet werden. Durch Widgets lassen sich verschiedenste Darstellungen wie verschiedene Diagramm-Typen, aber auch Texte und Tabellen darstellen. Dabei wird eine feste Darstellung auf die Daten vorkonfiguriert, welche später durch die Nutzenden verwendet werden kann. Innerhalb der Diagramme sind verschiedene Möglichkeiten zur Interaktion wie Filter (Sisense, o. J. - g) und Drilldowns (Deaggregationen) (Sisense, o. J. - c) möglich. Für Drilldowns können entweder Attribut-Hierarchien vorgegeben werden, die vom Nutzenden im Diagramm ausgewählt werden können und direkt die Visualisierung anpassen. Alternativ dazu, können auch bereits existierende Dashboards verlinkt werden, welche mit einem Filterwert aufgerufen werden können. Dadurch kann nach einem Klick in das Diagramm auch eine komplett andere Darstellung geladen werden (Sisense, o. J. - c) (Sisense, o. J. - a). Die jeweiligen Widgets können jedoch später nicht durch die Nutzenden frei angepasst werden (z. B. Anpassung des Diagramm-Typen).

Alle erzeugten Dashboards sind zunächst privat und können für andere Nutzende freigegeben werden. Nach einer Qualitätssicherung könnten Dashboards auch extern geteilt werden (Sisense, o. J. - i). Eine native Integration in Drupal zur Einbettung der Dashboards im UBA CMS existiert nicht. Es gibt jedoch eine JavaScript Bibliothek (Sisense, o. J. - e) zur Integration von Sisense in andere Anwendungen. Dadurch kann ein Drupal Modul implementiert werden, welches veröffentlichte Dashboards als Data-Outputs einbetten könnte. Das Erkunden von Datensätzen durch die Nutzenden der Webseite ist in der Standard-Funktionalität von Sisense nicht enthalten. Datensätze müssten daher entweder über mehrere Dashboards hinweg verlinkt werden, oder es müssten Dashboards mit verschiedenen Datensätzen zum Vergleich bereitgestellt werden. Sisense bietet zusätzlich auch die Möglichkeit eigene JavaScript Widgets zu implementieren, um Funktionalitäten zu ergänzen (Sisense, o. J. - j).

Sisense ist als Web-Anwendung entwickelt, die on-premise im UBA Rechenzentrum installiert werden kann.

Preise sind über die Webseite nicht bekannt und müssten über den Sisense-Vertrieb angefragt werden.

3.5.4.2 Anforderungstabelle

In der folgenden Tabelle werden alle Anforderungen an den Data Cube durch die Lösungskomponente beschrieben und nach der Klassifikation aus Kapitel 3.5 bewertet.

Tabelle 5: Anforderungstabelle der Lösungskomponente Sisense

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
AA1	Die Software soll erweiterbar/ anpassbar sein, falls nicht alle Anforderungen abgedeckt werden können (Anpassung von Design, aber auch Funktionalitäten)	o	Innerhalb von Dashboards können eigene JavaScript Erweiterungen implementiert werden. Bestehende Funktionalitäten können nicht angepasst werden.	https://sisense.dev/guides/js/extensions.html
AA2	Die Anforderungen müssen durch die Software effizient implementiert werden können.	+	Fertige Produktlösung, die mit den Standardfunktionen schnell verwendet werden kann. Um alle Anforderungen des Data Cube abzudecken, muss jedoch mit Einbettungen durch die JavaScript API und Konfiguration gerechnet werden.	
AA3	Die Software soll mit geringem Aufwand in Betrieb genommen werden.	+	Fertige Produktlösung, die on-premise installiert werden kann. Der eigene Implementierungsaufwand (Embedded Analytics) ist vergleichsweise gering gegenüber einer kompletten Eigenentwicklung.	
AA4	Die Software muss für Anwender und Administratoren ausreichend dokumentiert sein.	+	Gute Dokumentation sowohl für Anwender als auch für Entwickler.	https://documentation.sisense.com/ https://sisense.dev/guides/
AA5	Die Software ist bereits ausgereift und kann direkt verwendet werden.	+	Ausgereifte Produktlösung.	
AA6	Es gibt eine aktive Community mit Diskussionsforen	+	Es existiert ein Diskussionsforum.	https://community.sisense.com/t5/discussions/ct-p/discussion-forums

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	und Beispielen zur Anwendung.			
AA7	Die Software ist kostenfrei nutzbar, oder die Kosten sind in einem für das Projekt vertretbaren Rahmen.	-	Über die Webseite sind keine Kosten bekannt.	
AA8	Die Software kann im UBA Rechenzentrum betrieben werden	+	On-Premise oder Cloud Lösung möglich.	
AA9	Die Software kann ohne Weiteres gratis getestet werden	o	Als Testversion ist lediglich der Cloud-Zugang verfügbar.	
DO1	Es muss eine Visualisierung von Daten möglich sein, bei denen eine feste Ansicht auf die Daten und eine feste Darstellungsform durch die Redaktion vorgegeben wird.	+	Mit Sisense können Dashboards erzeugt werden, die feste Ansichten auf Daten ermöglichen.	https://documentation.sisense.com/8-8/creating-dashboards/dashboards.htm#gsc.tab=0
DO2	Daten müssen als Tabelle dargestellt werden können.	+	Tabellen sind als Dashboard-Widgets verfügbar.	https://documentation.sisense.com/8-8/creating-dashboards/adding-widgets-to-dash/table.htm#gsc.tab=0
DO3	Daten innerhalb der Tabellendarstellung sollen gefiltert werden können.	+	Alle Widgets können mit Filter versehen werden.	https://documentation.sisense.com/8-8/creating-dashboards/filtering-dashboards-and-widgets/introduction-to-filters.htm#gsc.tab=0
DO4	Daten innerhalb der Tabellendarstellung sollen sortiert werden können.		Funktionalität ist durch die Dokumentation nicht ersichtlich.	
DO5	Aggregationen von Datensätzen sollen vorgenommen oder vorberechnete	o	Funktionalität ist durch die Dokumentation nicht ersichtlich. Da Datenquellen über SQL angebunden werden können, wären vordefinierte	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Aggregationen angezeigt werden können.		Aggregationen möglich. Alternativ könnten Aggregationen im Data-Input erzeugt werden. Eine weitere Möglichkeit ist die Implementierung eigener JavaScript Widgets.	
DO6	Deaggregationen von Datensätzen sollen vorgenommen oder vorberechnete Deaggregationen angezeigt werden können.	+	Als drilldown möglich. Diese müssen zuvor bei der Erstellung von Widgets entsprechend konfiguriert werden. Dabei können entweder Attribut-Hierachien vorgegeben werden oder es können andere Dashboards für drilldowns mit entsprechenden Filtern verlinkt werden.	https://documentation.sisense.com/8-1/using-dashboards/drilling-down-widget.htm#gsc.tab=0 https://dtdocs.sisense.com/article/drilldowns
DO7	Zeitreihen sollen eingeschränkt werden können (Filterung durch min. und max. Datum).	+	Alle Widgets können mit Filter versehen werden. Filter-Widgets für Zeitreihen sind verfügbar.	https://documentation.sisense.com/8-8/creating-dashboards/filtering-dashboards-and-widgets/introduction-to-filters.htm#gsc.tab=0 https://documentation.sisense.com/8-1/creating-dashboards/working-with-time/time-in-sisense.htm#gsc.tab=0
DO8	Einheiten sollen dynamisch umgerechnet werden, oder durch vordefinierte Daten in anderen Einheiten angezeigt werden können.	o	Funktionalität ist durch die Dokumentation nicht ersichtlich. Bei Bedarf könnten Datensätze mit verschiedenen Einheiten durch den Data-Input abgelegt werden. Eine weitere Möglichkeit ist die Implementierung eigener JavaScript Widgets.	
DO9	Die Darstellungsform der Abbildungen soll durch den Nutzenden frei gewählt werden können. Gewünschte Darstellungsformen sind: Liniendiagramm, Balkendiagramm,	o	Die Darstellungsform kann nicht frei definiert werden. Folgende Typen können durch die Redaktion vorgegeben werden: Area Chart, Area Map, Bar Chart, Box Whisker Plot, Calendar Heatmap, Column Chart, Indicator, Line Chart, Pie Chart, Pivot, Polar Chart, Scatter Map, Sunburst, Text Widgets, Treemap	https://documentation.sisense.com/8-1/creating-dashboards/adding-widgets-to-dash/add-widget-to-dash.htm#gsc.tab=0

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Tortendiagramm, Baumdiagramme, Abweichungen, Korrelationen und Streudiagramme, Häufigkeitsverteilungen, Nominale Vergleiche (wie Blasendiagramme oder Heatmaps), Fluss und Sankey-Diagramme, Netzwerkdiagramme)			
DO10	Die Visualisierungen in Diagrammen/Abbildungen (Data-Outputs) müssen direkt in Artikel auf der UBA-Webseite (Drupal CMS) eingebunden werden können	o	Es existiert keine native Drupal Integration. Zur Integration in andere Anwendungen sind jedoch mehrere API Varianten oder iFrames vorgesehen.	https://sisense.dev/guides/embedding/
DO11	Die Visualisierungen in Diagrammen/Abbildungen (Data-Outputs) soll in der zukünftigen umwelt.info Webseite möglich sein.	o	Es existiert keine native Webseiten-Integration (z.B. Drupal) ohne iFrames. Zur Integration in andere Anwendungen sind jedoch mehrere API Varianten vorgesehen.	https://sisense.dev/guides/embedding/
DO12	Die Implementierung soll möglichst CMS-offen (unabhängig von dem bestehenden Drupal System) erfolgen. Es soll geprüft werden ob Inhalte z.B. als reponsive iFrames oder vergleichbares eingebettet werden können.	+	Die Dashboards selbst sind CMS neutral implementiert. Lediglich für die Integration ist eine Drupal-spezifische Lösung notwendig, falls keine iFrames zum Einsatz kommen sollen.	
DO13	Funktionalitäten die nicht CMS-offen	+	Drupal Standards können bei einer Drupal-Modul-	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	implementiert werden, sind als Drupal-Modul (unter Beachtung der Drupal Code-Konventionen) zu entwickeln in vollständig lauffähig in das CMS der UBA Webseite zu integrieren.		Implementierung berücksichtigt werden.	
DO14	Es muss möglich sein, Dashboards im Drupal CMS einzubinden, die aus mehreren Data-Outputs bestehen	+	Alle Data-Outputs von Sisense sind Dashboards. Es wird keine Unterscheidung zwischen Data-Output und Dashboard vorgenommen.	
DO15	Dashboards und Data-Outputs sollen auf verschiedenen Display Größen (Mobilgeräte und größere Bildschirme) gut angezeigt werden.	+	Sisense ist responsiv. Für eigene Drupal-Implementierungen müsste ebenfalls auf Responsivität geachtet werden.	
DO16	Daten und Abbildungen sollen in unterschiedlichen Formaten zum Download angeboten werden. Daten: CSV, Excel Abbildungen: PNG/JPEG	o	Download von Widgets als CSV möglich.	https://documentation.sisense.com/latest/using-dashboards/export-widget-csv.htm
DO17	Datensätze die zu einem Thema gehören sollen erkundet werden können. Das heißt, dass die Daten sowie verlinkte Datensätze angezeigt werden.	-	Keine Standard-Funktionalität. Könnte ggf. durch eigene JavaScript Widgets erweitert werden.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DO18	Verknüpfte Datensätze sollen in der Visualisierung mit hinzugefügt werden können, um Daten vergleichen zu können.	-	Es existiert keine Standard-Funktionalität. Verknüpfte Datensätze könnten nur durch vordefinierte Dashboards zeitgleich dargestellt werden. Könnte ggf. durch eigene JavaScript Widgets erweitert werden.	
DO19	Das Corporate Design ist bei der Gestaltung aller Nutzeroberflächen zu beachten.	+	Es existieren einfache Möglichkeiten zum Anpassen von Farben und Logos ("white labeling").	https://documentation.sisense.com/7-1/administration/embedded-analytics/rebranding-sisense/white-label.htm
DO20	Es ist zu berücksichtigen wie Data-Outputs auch in die Datensuche des UBA eingebunden werden können.	o	Wenn Data-Outputs als Drupal-Module implementiert werden, kann der Standard Drupal-Mechanismus verwendet werden. Hierzu müssen beim Erstellen der Inhalte in Drupal entsprechende Keywords vergeben werden.	
DE1	Über den Data-Explorer müssen Datensätze des Data Stores auffindbar gemacht werden.	+	Ist über die Datensuche möglich.	https://documentation.sisense.com/7-1/managing-data/working-with-data/searching-tables-fields.htm
DE2	Metadaten von Datensätzen sollen angezeigt werden können.	-	Über die Dokumentation nicht ersichtlich.	
DE3	Datensätze sollen als interaktive Diagramme angezeigt werden können.	+	Für die Anzeige muss zunächst ein Dashboard erzeugt werden. Die folgenden Typen sind in Dashboards möglich: Area Chart, Area Map, Bar Chart, Box Whisker Plot, Calendar Heatmap, Column Chart, Indicator, Line Chart, Pie Chart, Pivot, Polar Chart, Scatter Map, Sunburst, Text Widgets, Treemap	https://documentation.sisense.com/8-1/creating-dashboards/adding-widgets-to-dash/add-widget-to-dash.htm
DE4	Datensätze sollen tabellarisch angezeigt werden können.	+	Als Preview in der Datensuche oder in Dashboards möglich.	https://documentation.sisense.com/7-1/managing-data/working-with-data/preview-data.htm

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DE5	Eine Definition von zusammengehörigen Datensätzen für einen Cube soll möglich sein.	+	Datensätze können als ElastiCubes zusammengestellt werden. Inwieweit diese Informationen für externe Nutzende zur Exploration verwendet werden können, ist unklar.	https://documentation.sisense.com/7-1/managing-data/ElastiCubeManager.htm
DE6	Es soll möglich sein, alle Cubes aufzulisten.	+	Über den ElastiCube Manager möglich.	https://documentation.sisense.com/7-1/managing-data/ElastiCubeManager.htm
DE7	Es soll möglich sein alle Datensätze aufzulisten, die in einem Cube verwendet werden.	+	Über den ElastiCube Manager möglich.	https://documentation.sisense.com/7-1/managing-data/ElastiCubeManager.htm
DE8	Es soll möglich sein für einen Datensatz anzuzeigen, in welchem Cube er verwendet wird.	-	Aus der Dokumentation nicht ersichtlich.	
DE9	Für jeden Datensatz sollen fachlich begründete Dimensionen ausgewählt werden können.	+	Aus der Dokumentation ist nicht ersichtlich, ob dies direkt in der GUI möglich ist. Alternativ können neue Datensätze definiert werden, die entsprechend vorgefiltert werden können.	https://documentation.sisense.com/7-1/managing-data/transforming-data/add-custom-table.htm
DE10	Hierarchische Datensätze sollen miteinander verknüpft werden können.	+	Über den ElastiCube Manager möglich.	https://documentation.sisense.com/7-1/managing-data/ElastiCubeManager.htm
DE11	Datenauszügen für einzelne Datensätze sollen definiert werden können.	+	Aus der Dokumentation ist nicht ersichtlich, ob dies direkt in der GUI möglich ist. Alternativ können neue Datensätze definiert werden, die entsprechend vorgefiltert werden können.	https://documentation.sisense.com/7-1/managing-data/transforming-data/add-custom-table.htm
DE12	Eine Konfiguration von Datensätzen / Dimensionen für einzelne Data-Outputs soll ermöglicht werden. Es soll	o	Daten können nicht dynamisch hinzugeladen werden. Es ist jedoch möglich, vor der Veröffentlichung gemeinsame Dashboards für die potenziell zusammen anzuzeigenden Daten vorzukonfigurieren.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	konfigurierbar sein, welche weiteren Datensätze zu einem Data-Output potenziell hinzugeladen werden dürfen.			
DE13	Konfiguration der initialen und konfigurierbaren Darstellungsform	+	Standardfunktionalität der Dashboards	
DE14	Konfiguration weiterer Optionen des Data-Outputs für die Nutzenden: - Daten hinzuladen - Einheiten umrechnen / Datensatz mit anderen Einheiten laden - Aggregationen berechnen / Datensätze mit anderen Aggregationen laden - Erlaubte Downloads	o	Dashboards können vielfältig vordefiniert werden. Eine Anpassung durch den Nutzenden ist jedoch nicht mehr möglich. Ggf. sind einzelne Funktionen durch eigene JavaScript Widgets möglich.	
DE15	Für jeden Data-Output sollen die Zusatz-Funktionalitäten (z.B. Aggregationen berechnen, Daten hinzuladen, …) bei Bedarf deaktiviert werden können.	o	(De-)Aggregationen können pro Darstellung konfiguriert werden. Hinzuladen von Daten oder Umrechnen von Einheiten ist nicht konfigurierbar.	
DE16	Konfiguration von Dashboards (Kombination aus mehreren Data-Outputs)	+	Standardfunktionalität der Dashboards	
DE17	Der Data-Explorer soll außerhalb des UBA CMS implementiert werden können	+	Sisense ist eine eigenständige Anwendung.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	und hat kein spezifisches CMS zur Voraussetzung.			
DE18	Für externe Analysetools soll der Zugriff auf die Daten über eine API ermöglicht werden.	o	Es sind verschiedene APIs zum Abfragen von Daten verfügbar. Es ist unklar, ob die API auch zum Abfragen von Daten für externe Nutzende sinnvoll nutzbar ist.	https://sisense.dev/guides/
DE19	Für externe Analysetools soll der Zugriff auf die Daten über einen Export ermöglicht werden.	+	Daten aus Dashboards und Widgets können als Download bereitgestellt werden.	https://documentation.sisense.com/8-8/reporting/download-dash-widget-overview.htm#gsc.tab=0
DE20	Die Konfiguration der Data-Outputs und Data Cubes soll über ein Webinterface möglich sein.	+	Sisense ist eine Web-Anwendung.	
DE21	Es soll möglich sein Zusatzdateien für Berichte (Bilder in verschiedenen Formaten, Daten) herunterzuladen.	o	Im Standard nicht möglich. Könnte ggf. über die APIs implementiert werden.	
DE22	Änderungen an den Daten sollen angezeigt werden können (Überarbeitungsmodus)	-	Aus der Dokumentation nicht ersichtlich.	
DI1	Es soll eine Nutzeroberfläche und ein Workflow gestaltet werden um weitere Datenquellen anzubinden.	+	Ist über den Data Manger möglich.	https://documentation.sisense.com/8-8/managing-data/IntroToWorkData.htm#gsc.tab=0
DI2	Der Data-Input soll entweder durch die bestehenden Excel Templates der Redaktion oder falls möglich durch eine direkte Anbindung der Datenhaltung der		Es sind verschiedene „Connnectoren" vorhanden, um diverse standardisierte und proprietäre Datenbanken und Dokumentformate anzubinden. Der Data-Input ist grundsätzlich jedoch unabhängig von Sisense.	https://documentation.sisense.com/docs/introduction-to-data-sources

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	datenhaltenden Stellen durchgeführt werden können.			
DI3	Daten sollen beim Import eine Qualitätssicherung durchlaufen		Unabhängig von Sisense.	
DI4	Der Data-Input muss mit verschiedenen Eingangsformaten (Datenbanken, Excel, CSV, WebServices) umgehen können um Daten von verschiedenen datenhaltenden Stellen einzulesen.	+	Verschiedene Datenquellen sind möglich (Datenbanken, online cloud storages, CSV, …). Eine Liste der möglichen Datenhaltungsquellen ist im Quell-Link zu finden. Siehe auch DI2. Andere Datenquellen müssten durch ein ETL-Tool zunächst aufbereitet werden.	https://documentation.sisense.com/docs/introduction-to-data-sources
DI5	Metadaten, welche in den Datenquellen enthalten sind sollen mit ausgelesen werden.		Unabhängig von Sisense.	
DI6	Während des Data-Inputs sollen zusätzliche Metadaten für einen Datensatz angereichert werden können.		Unabhängig von Sisense.	
DI7	Während des Data-Inputs sollen Verlinkungen zu anderen Datensätzen im Data-Store möglich sein.	+	Kann ggf. durch eine eigene Datenstruktur abgebildet werden. Es könnte jedoch auch die GUI Funktionalität im ElastiCube Manager verwendet werden.	
DI8	Mit dem Data-Input soll es möglich sein, mehrere Versionen von einem Datensatz einzulesen und im		Unabhängig von Sisense.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Data-Store zu speichern.			
DI9	Für Datensätze mit gleichbleibenden Strukturen soll ein automatisierter Importprozess möglich sein.		Unabhängig von Sisense.	
DI10	Für Automatisierungen soll es möglich sein, zeitliche Ablaufpläne zu konfigurieren, um Datensätze regelmäßig zu importieren.		Unabhängig von Sisense.	
RP1	Es muss ein redaktioneller Prozess zur Pflege der Daten zur Umwelt entwickelt werden, oder der DataCube muss in den bestehenden Prozess mit eingebettet werden.	+	Der redaktionelle Prozess ist grundsätzlich unabhängig von Sisense. Da jedes in Sisense erstelle Dashboard zunächst privat ist und erst veröffentlich werden muss, können viele Anforderungen an QS und Freigaben einfach umgesetzt werden.	
RP2	Daten dürfen nur nach einer Überprüfung durch einen Data-Output auf der Webseite dargestellt werden.	+	Dashboards müssen erst freigegeben werden, bevor diese durch andere Nutzende einsehbar sind.	https://documentation.sisense.com/latest/managing-dashboards/share-dashboard.htm
RP3	Data-Outputs müssen aufwandsarm in Artikel der UBA Webseite eingebettet werden können.	+	Verschiedene Einbettungsmöglichkeiten sind möglich (vgl. DO10, DO11)	
RP4	Data-Outputs dürfen nur nach vorheriger Freigabe veröffentlicht werden.	+	Dashboards müssen erst freigegeben werden, bevor diese durch andere Nutzende einsehbar sind.	https://documentation.sisense.com/latest/managing-dashboards/share-dashboard.htm

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
RP5	Bei Anpassung der Daten eines Berichts/Artikels sollen Redakteure benachrichtigt werden.	o	Unabhängig von Sisense. Muss im Data-Input implementiert werden.	
RP6	Der manuelle Aufwand des redaktionellen Prozesses soll soweit möglich reduziert werden.		Unabhängig von Sisense.	
DS1	Es ist eine zentrale Datenhaltung mit allen Daten zur Umwelt aufzubauen.	+	Innerhalb von Sisense wird im Regelfall keine interne Datenhaltung aufgebaut. Es können verschiedene (dezentrale) Datenhaltungsquellen verwendet und ebenso gleichzeitig angebunden werden. Verschiedene Datenquellen können durch ElastiCubes in Sisense neu strukturiert werden, weshalb kein Datenmodell vorgeben werden muss. Es können daher unterschiedlichste Datenmodelle verwendet werden.	
DS2	Die Datenstrukturen müssen klar definiert werden. Es muss ein themenübergreifender Gesamtansatz für das Datenmodell ausgearbeitet werden, in den sich die verschiedenartigen Daten zur Umwelt einordnen.	+	s. DS1 und DS2	
DS3	Das Datenmodell muss so konzipiert werden, dass es einerseits beliebig vertieft/ detailliert	o	s. DS1 und DS2	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	und andererseits beliebig fachlich erweitert werden kann.			
DS4	Die datenhaltenden Stellen sollen ihre Daten möglichst automatisiert in den Data-Store einspielen können.		Unabhängig von Sisense.	
DS5	Auch rückwirkend müssen Datenaktualisierungen und -erfassungen im Data-Store an die Fachsysteme (teil-) automatisiert übergeben werden können.		Unabhängig von Sisense.	
DS6	Die Webseite soll Daten direkt aus dem Data-Store nutzen.	+	Ist durch die direkte Anbindung der Dashboards an die Datenhaltung gegeben.	
DS7	Datenänderungen sollen automatisch an die Redakteure bzw. Fachexperten/-innen übermittelt werden.		Unabhängig von Sisense.	
DS8	Die Daten müssen über Status so markiert sein, dass sofort erkennbar ist, in welchem Bearbeitungszustand sie sich gerade befinden.		Unabhängig von Sisense.	
DS9	Dynamische Zugriffe auf die Daten müssen in Abhängigkeit des Status definiert und kontrolliert werden können.	o	Wenn Status-Informationen für alle Datensätze gepflegt werden, können diese in Dashboard Filtern entsprechend verwendet werden.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DS10	Es muss möglich sein, Daten aus verschiedenen Quellsystemen (insbesondere aus DESTATIS) zu übernehmen, diese einheitlich strukturiert zu verwalten, sodass sie dann als Dimensionen der unterschiedlichen Daten zur Umwelt im Data-Store genutzt werden können.		Unabhängig von Sisense.	
DS11	Es muss möglich sein, für alle erfassten Datensätze Dimensionen zu definieren.		Unabhängig von Sisense.	
DS12	Es muss möglich sein, die übernommenen Daten (konkrete Werte der Datensätze) den entsprechenden Dimensionen zuzuordnen.		Unabhängig von Sisense.	
DS13	Die Werte müssen auch als Werte (Zahlen, Vektoren) gespeichert werden, sodass mit ihnen gerechnet werden kann.		Unabhängig von Sisense.	
DS14	Es soll möglich sein, im Data-Store Berechnungen zu hinterlegen, die automatisch ausgeführt werden.	+	Es können "Custom Fields" an Datensätze ergänzt werden, in denen Berechnungen durchgeführt werden können.	https://documentation.sisense.com/latest/managing-data/transforming-data/add-custom-field.htm#gsc.tab=0
DS15	Es soll möglich sein, bestimmte Berechnungen	+	Es können "Custom Fields" an Datensätze ergänzt werden, in	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	manuell anzustoßen.		denen Berechnungen durchgeführt werden können.	
DS16	Es soll möglich sein, Berechnungen hierarchisch auszuführen. Das gilt auch für automatisierte Berechnungen.		Unabhängig von Sisense.	
DS17	Es muss auch weiterhin möglich sein, extern mit den Daten zur Umwelt zu arbeiten und externe Daten mit ihren Bezügen zu den Ausgangsdaten im Data-Store einzuspielen.	-	Unabhängig von Sisense. Es müsste jedoch ein Export/Import Workflow zur Bearbeitung von Daten außerhalb des Tools etabliert werden. Dies könnte z.B. durch ETL-Prozesse realisiert werden.	
DS18	Die Werte müssen historisch verwaltet werden.		Unabhängig von Sisense.	
DS19	Bei hierarchischen Berechnungen müssen die Bezüge der Daten jederzeit wieder selektierbar sein.		s. DS18	
DS20	Objektklassen, die für die Dimensionierung herangezogen werden, müssen ebenfalls historisch verwaltet werden, um ältere Daten zur Umwelt auch weiterhin einordnen zu können.		s. DS18	

3.5.4.3 Bewertung

Die Lösungskomponente Sisense ist eine etablierte Software, die mit einem großen Fokus auf „Embedded-Analytics“ konzipiert wurde. Das bedeutet, dass verschiedenste APIs bereitstehen, um die Anwendung in andere Lösungen zu integrieren. Diese sind gut dokumentiert und würden eine gute Integration in die UBA-Webseite ermöglichen. Bei Bedarf können zusätzlich zumindest einfache Anpassungen am Design vorgenommen werden, um Sisense an das UBA-Design anzupassen. Innerhalb der Dashboards können eigene Widgets implementiert werden, um auch eigene Funktionalitäten bereitzustellen. Für eine Integration in das UBA-CMS muss jedoch eine eigene Drupal-Modul-Implementierung auf Basis der APIs durchgeführt werden.

Zur Anbindung an Daten sind eine Vielzahl möglicher Datenquellen über eine grafische Benutzeroberfläche anbindbar, welche zusätzlich zu ElastiCubes verbunden werden können, wodurch eine flexible Datenhaltung möglich ist. Innerhalb der Dashboards können viele verschiedene Widgets verwendet werden, um zum Beispiel verschiedene Diagramm-Typen und Tabellen abzubilden. Für die Data-Explorer Komponente lassen sich daher viele Anforderungen abbilden.

Sisense wurde vor allem für Business Intelligence Analysen durch interne Mitarbeitende konzipiert. Mit Bezug auf den Data-Output werden einige Funktionen daher nur teilweise abgedeckt. Während fest konfigurierte Darstellungen sehr gut möglich sind, sind wenig Konfigurationsmöglichkeiten durch externe Nutzende möglich. Nachträgliche Anpassung von Diagramm-Typen oder das Hinzuladen von Datensätzen sind nicht vorgesehen. Auch das Erkunden von verlinkten Datensätzen ist im Standard nicht enthalten. Es sind verschiedene Ansätze denkbar, wie eine ähnliche Funktionalität bereitgestellt werden könnte, welche jedoch stets einen hohen manuellen Konfigurationsaufwand in den Dashboards bedeuten würden.

Sisense ist eine kommerzielle Software. Über die Webseite sind keine Kosten für Sisense bekannt, weshalb für eine genaue Kostenanalyse der Sisense-Vertrieb kontaktiert werden muss. Es muss mit hohen Lizenzkosten gerechnet werden.

Als fertige Software-Lösung wird davon ausgegangen, dass eine Anbindung an die eigene Datenhaltung sowie erste Visualisierungen mit geringem Zeitaufwand erstellt werden können, da diese durch die Standard-Funktionalität von Sisense bereits abgedeckt werden. Für die Drupal-Integration sowie die Erkundungsmöglichkeiten durch externe Nutzende muss mit eigenem Implementierungsaufwand gerechnet werden.

3.5.5 Tableau

3.5.5.1 Lösungsansatz

Tableau ist eine kommerzielle Softwarelösung zur Datenanalyse und Visualisierung, welche aus verschiedenen Komponenten besteht und die durch diverse Add-ons erweitert werden kann. Im Folgenden wird der gesamte Prozess des Data Cube Prozesses von Datenverarbeitung über Datenaufbereitung bis zur Visualisierung und Erkundung der Daten skizziert, und die in dem jeweiligen Kontext benötigten Tableau Komponenten werden kurz erläutert.

Am Anfang des Prozesses steht die Daten Vor- und Aufbereitung im Mittelpunkt. Tableau Prep bietet hier Möglichkeiten, die Anforderungen an die Data-Input Komponente abzubilden. In den zwei Teilprodukten Tableau Prep Builder und Tableau Prep Conductor stehen jeweils unterschiedliche Schwerpunkte im Fokus: Ersteres kann für die Aufbereitung der Daten hinsichtlich Bereinigungsschritte, das Zusammenführen verschiedener Datenquellen und Transformationsprozesse genutzt werden. An dieser Stelle ist auch die Verwendung einer alternativen ETL-Softwarelösung denkbar. Der Tableau Prep Conductor wird im Rahmen des

Data Management Add-on ausgeliefert und unterstützt die Automatisierung der Abläufe zur Datenvorbereitung. Hierzu zählen zum Beispiel das Einplanen von Daten-Update Prozessen genauso wie die Überwachung dieser Abläufe, die bei Tableau mit dem Begriff Schemata beschrieben werden. Für ergänzende Details zu der Tableau Prep Komponente wird auf die zugehörige Website (Tableau Software, LLC, o. J. - k) verwiesen.

Während die Datenhaltungskomponente generell als unabhängig von Tableau angesehen werden kann, wird dennoch darauf hingewiesen, dass Tableau die Anbindung an zahlreiche verschiedene Datenquellen ermöglicht. Hierfür seien repräsentativ Oracle-, MySQL- und PostgreSQL-Datenbanken genannt. Zudem werden auch Microsoft Excel Dateien als Datenquelle unterstützt (Tableau Software, LLC, o. J. - l). Somit kann die Data-Store Komponente grundsätzlich losgelöst von Tableau realisiert werden, solange die Datenhaltungskomponenten von Tableau unterstützt werden. In diesem Zusammenhang wird weiterhin angemerkt, dass es als empfehlenswert erscheint, relationale Datenbanken zu verwenden. Prinzipiell ist die Anbindung an die Cube-Datenquellen „Oracle Essbase, Teradata OLAP, Microsoft Analysis Services (MSAS), SAP NetWeaver Business Warehouse [und] Microsoft PowerPivot“ (Tableau Software, LLC, o. J. - a) möglich, jedoch stehen im Falle von Cube-Datenquellen nicht alle Tableau-Funktionalitäten zur Verfügung. Laut der Online-Hilfe gibt es alternative Ansätze, um die „Nichtverfügbarkeit dieser Funktionen zu kompensieren“ (Tableau Software, LLC, o. J. - a). Falls die Wahl auf Tableau fallen sollte, wird empfohlen, die Anbindung an Cube-Datenquellen nähergehend zu eruieren und gegenüber relationalen Datenbanken abzuwägen. In diesem Zusammenhang wird darauf hingewiesen, dass insbesondere das Stern- und das Schneeflockenschema für die Datenmodellierung in mehrdimensionalen Datenräumen von Tableau unterstützt werden (Tableau Software, LLC, o. J. - b).

Nachdem der Data-Input und der Data-Store behandelt wurden, wird im Folgenden davon ausgegangen, dass die Daten nun zur weiteren Analyse mit dem Data-Explorer bereitstehen. Für diese Komponente des Data Cube erscheint der Einsatz von sowohl Tableau Desktop als auch Tableau Server als sinnvoll. Die folgenden Beschreibungen basieren auf den Online-Dokumentation für Tableau Desktop (Tableau Software, LLC, o. J. - j) und Tableau Server (Tableau Software, LLC, o. J. - m).

Tableau Desktop ist im Kontext des Data Cube ein wichtiges Werkzeug für die Datenanalyse. Mit diesem können verschiedenste Visualisierungen und Ansichten erstellt werden. Sowohl die Darstellung als Tabelle als auch die Verwendung von unterschiedlichsten Diagrammen und Graphen wird unterstützt. Für die Datenexploration können Filter- und Sortierfunktionen genutzt werden und ebenso Aggregationen durchgeführt werden. Hierfür werden Berechnungen wie z. B. das Aufsummieren standardmäßig bereitgestellt. Zusätzlich zu den vordefinierten Berechnungen können eigene komplexe Berechnungsvorschriften implementiert werden. In diesem Kontext ist es weiterhin wichtig, dass nicht nur ein Datensatz bzw. eine Datenquelle für die Analyse und Generierung von Visualisierungen genutzt werden kann, sondern vielmehr auch unterschiedliche Daten(quellen) kombiniert und zusammengeführt werden können. Tableau nutzt hierzu unter anderem eine eigene proprietäre Datenbank namens Hyper, welche neu zusammengestellte Daten noch einmal speziell aufbereitet vorhalten kann (Tableau Software, LLC, o J. - d).

Mit Tableau Desktop können Data-Outputs generiert werden, die dann im weiteren Verlauf veröffentlicht werden können. Tableau Desktop wird in diesem Sinne als internes Werkzeug gesehen, um interaktive Diagramme und Dashboards herzustellen, die dann anderen Mitarbeitenden zur Kollaboration und Erkundung bereitgestellt werden können. Ebenso ist die Einbettung auf der UBA-Webseite möglich, um freigegebene Data-Outputs der Öffentlichkeit zur Erkundung bereitzustellen. Hierfür wird eine spezielle Lizensierung, eine sogenannte kern-

basierte Lizenz, benötigt, um den Gastzugriff zu realisieren (Tableau Software, LLC, o. J. - h) (Tableau Software, LLC, o. J. - f). In diesem Kontext wird darauf hingewiesen, dass die Einbettung in Drupal grundsätzlich denkbar ist. Für Drupal 8 ist das Modul „Tableau Dashboard Integration“ frei verfügbar (Avbar, Norton, & Chadwick, o. J.), jedoch ist die Unterstützung für Drupal 9 derzeit unklar. Möglicherweise müsste hier ein eigenes Drupal-Modul zur Einbettung der Tableau-Dashboards implementiert werden. An dieser Stelle kann die Verwendung Tableau JavaScript API hilfreich sein, um „Tableau-JavaScript-Objekte in Webanwendungen“ zu verwenden (Tableau Software, LLC, o. J. - c). Ebenso könnte es zielführend sein, den Einbettungscode von Tableau für Ansichten zu verwenden (Tableau Software, LLC, o. J. - c).

Mit der Tableau Server Komponente können Datenquellen, Visualisierung und Dashboards verwaltet und veröffentlicht werden. In diesem Sinn bedeutet „Veröffentlichung“ prinzipiell erst einmal die Freigabe und das Teilen von Daten und Inhalten zur Erkundung für andere Nutzende und nicht direkt die Veröffentlichung von Data-Outputs online auf der UBA-Webseite. Um auch externen Nutzenden „das Anzeigen und die Interaktion mit Tableau-Ansichten“ zu ermöglichen, wird empfohlen für die Ansichten „URL-Links bereit[zustellen]“ oder diese direkt in Webseiten einzubetten (Tableau Software, LLC, o. J. - f). Zusätzlich zur Exploration der Ansichten ist es möglich, dem Gastnutzenden das „Anzeigen und Herunterladen“ der Datenquelle zu erlauben (Tableau Software, LLC, o. J. - f). Weiterführende Informationen zur Freigabe von Webinhalten sind auf der zugehörigen Seite der Tableau Online-Hilfe zu finden (Tableau Software, LLC, o. J. - e).

Die Verwaltung von Nutzenden und die Pflege von Berechtigungsstufen auf Nutzenden- und Gruppenebene erfolgt ebenfalls in der Tableau Server Umgebung. Somit können für unterschiedliche interne Mitarbeitende verschiedene Funktionalitäten bereitgestellt werden, so dass zum Beispiel einige Mitarbeitende nur freigegebene Inhalte erkunden dürfen, wohingegen andere neue Visualisierungen und Dashboards erstellen und gegebenenfalls sogar freigeben können. Tableau Server ermöglicht zudem die grafische Visualisierung und Überwachung der Nutzung von Datenquellen und Inhalten. Ebenso können Administrierende mit diesem Werkzeug Tableau Lizenzen verfolgen und verwalten.

Für den Redaktionellen Prozess wird die Anwendung verschiedener Tableau Komponenten gesehen. Um den Prozess, wie in Kapitel 3.4 beschrieben, abzudecken, sollte zunächst Tableau Prep verwendet werden, um den Data-Input abzubilden. Die Komponenten Tableau Desktop und Tableau Server werden verwendet, sowohl um Data-Outputs zu generieren als auch um diese mit anderen zu teilen, auszutauschen und schließlich zu veröffentlichen. Da diese Teilschritte im Einzelnen bereits erläutert wurden, werden sie im Rahmen des Redaktionellen Prozesses nicht erneut tiefergehend beschrieben.

3.5.5.2 Anforderungstabelle

In der folgenden Tabelle werden alle Anforderungen an den Data Cube durch die Lösungskomponente beschrieben und nach der Klassifikation aus Kapitel 3.5 bewertet.

Tabelle 6: Anforderungstabelle der Lösungskomponente Tableau

Nummer	Beschreibung	Bewertung	Umsetzung	Quelle
AA1	Die Software soll erweiterbar/ anpassbar sein, falls nicht alle Anforderungen	o	Die Komponente "Embedded Analytics" bietet Schnittstellen-Funktionen (z. B. JavaScript API, Rest API) zum Einbetten und Anpassen von Tableau. Damit	https://www.tableau.com/de-de/embedded-analytics

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	abgedeckt werden können (Anpassung von Design, aber auch Funktionalitäten)		kann z. B. das UBA Corporate Design berücksichtigt werden ("white labeling"). Die Software selbst kann nicht angepasst werden.	
AA2	Die Anforderungen müssen durch die Software effizient implementiert werden können.	+	Fertige Produktlösung, die mit den Standardfunktionen schnell verwendet werden kann. Um alle Anforderungen des Data Cube abzudecken, muss jedoch mit Einbettungen durch die JavaScript API und Konfiguration gerechnet werden.	https://www.tableau.com/de-de/why-tableau
AA3	Die Software soll mit geringem Aufwand in Betrieb genommen werden.	+	Die Installation und Konfiguration von mehreren Tableau-Komponenten (Tableau Server, Tableau Desktop, Embedded Analytics, Tableau Prep oder FME, ggf. Data Management Add-On) ist für die Inbetriebnahme notwendig. Der eigene Implementierungsaufwand (Embedded Analytics) ist vergleichsweise gering gegenüber einer kompletten Eigenentwicklung.	
AA4	Die Software muss für Anwender und Administratoren ausreichend dokumentiert sein.	+	Es gibt Online-Dokumentation wie z.B. die Knowledgebase, Tableau-Hilfe (auch als PDF verfügbar), Training Videos und weitere (online) Schulungsmöglichkeiten.	https://www.tableau.com/de-de/support
AA5	Die Software ist bereits ausgereift und kann direkt verwendet werden.	+	Laut der Webseite ist Tableau "Markführer" für moderne BI Lösungen.	https://www.tableau.com/de-de/why-tableau
AA6	Es gibt eine aktive Community mit Diskussionsforen und Beispielen zur Anwendung.	+	Es ist keine open-source Lösung. Die Tableau Community besteht aus mehr als einer Millionen Mitgliedern weltweit. Es existiert eine große Fragen- und Ideensammlung mit mehr als 195.000 Beiträgen. Die Community ist aktiv, was z. B. der aktuelle Blog-Post vom	https://www.tableau.com/de-de/community

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			10.11.2021 oder der Forums-Beitrag von Oktober 2021 zeigt (abgerufen 11.11.2012).	
AA7	Die Software ist kostenfrei nutzbar, oder die Kosten sind in einem für das Projekt vertretbaren Rahmen.	-	Für den Data Cube muss voraussichtlich entweder eine Tableau Core oder eine Tableau Embedded Lizenz verwendet werden, für welche keine Kosten auf der Webseite angegeben sind. Es sind jedoch hohe Kosten zu erwarten. Für eine rein interne Nutzung könnte eine Nutzerbasierte Lizenzierung verwendet werden.	https://www.tableau.com/de-de/pricing/teams-orgs https://help.tableau.com/current/server/de-de/processes.htm#LicensedProcess
AA8	Die Software kann im UBA Rechenzentrum betrieben werden	+	Die Software kann im UBA Rechenzentrum installiert werden (on-premise Lösung).	
AA9	Die Software kann ohne Weiteres gratis getestet werden	+	Es gibt kostenlose Testversionen z. B. für Tableau Desktop, Tableau Server und Tableau Prep Builder.	https://www.tableau.com/de-de/products/trial
DO1	Es muss eine Visualisierung von Daten möglich sein, bei denen eine feste Ansicht auf die Daten und eine feste Darstellungsform durch die Redaktion vorgegeben wird.	+	Mit Tableau können Tableau-Ansichten und -Dashboards erstellt und veröffentlicht werden (Tableau Desktop und Tableau Server). Hierbei sollte die gewünschte Funktionalität (oder auch fehlende Interaktivität) vor der Veröffentlichung bei der Erstellung mit berücksichtigt werden.	
DO2	Daten müssen als Tabelle dargestellt werden können.	+	Es werden Tabellenansichten und -strukturen unterstützt.	https://help.tableau.com/current/pro/desktop/de-de/buildmanual_dragging.htm
DO3	Daten innerhalb der Tabellendarstellung sollen gefiltert werden können.	+	Das Filtern auf bestimmte Bereiche wie z.B. Kategorien oder Daten ist möglich.	https://help.tableau.com/current/pro/desktop/de-de/filtering.htm https://help.tableau.com/current/pro/desktop/de-de/actions.htm https://help.tableau.com

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
				/current/reader/desktop/de-de/interact.htm
DO4	Daten innerhalb der Tabellendarstellung sollen sortiert werden können.	+	Tabellen können alphabetisch und numerisch sortiert werden.	https://help.tableau.com/current/pro/desktop/de-de/sortgroup_sorting_computed_howto.htm https://help.tableau.com/current/reader/desktop/de-de/interact.htm
DO5	Aggregationen von Datensätzen sollen vorgenommen oder vorberechnete Aggregationen angezeigt werden können.	+	Seitens Tableau vordefinierte und individuell neu definierte Aggregationen (Drill Up) von Kennzahlen sind genauso möglich wie Drill Down Funktionen (Deaggregationen). Diese werden in Tableau Desktop vor der Veröffentlichung festgelegt, um so dem Nutzenden der Webseite bestimmte (De-)Aggregationsmöglichkeiten bereitzustellen. Hierbei können Standardaggregationen vor der Veröffentlichung definiert werden, die nicht durch den Nutzenden verändert werden können. Wichtig: Mehrdimensionale Datenquellen werden nur unter Windows unterstützt. Bei der Verwendung von Cube-Datenquellen entfallen ggf. einige Funktionen wie z.B. bestimmte Aggregationsfunktionen zum Bilden von Summen und Mittelwerten. Diese können jedoch häufig selbst nachgestellt werden. Im besten Fall kann eine relationale Datenbank verwendet werden, die gleichzeitig als Quelle für die Cube-Datenquelle verwendet wurde.	https://help.tableau.com/current/pro/desktop/de-de/calculations_aggregation.htm https://help.tableau.com/current/pro/desktop/de-de/buildmanual_multidimensional_drilldown.htm https://help.tableau.com/current/pro/desktop/de-de/cubes.htm
DO6	Deaggregationen von Datensätzen sollen vorgenommen oder	+	siehe DO5	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	vorberechnete Deaggregationen angezeigt werden können.			
DO7	Zeitreihen sollen eingeschränkt werden können (Filterung durch min. und max. Datum).	+	Tableau unterstützt relative Datumsfilter sowie das Filtern auf eine bestimmte Zeitspanne.	https://help.tableau.com/current/pro/desktop/de-de/qs_relative_dates.htm
DO8	Einheiten sollen dynamisch umgerechnet werden, oder durch vordefinierte Daten in anderen Einheiten angezeigt werden können.	o	Es ist möglich, Werte in andere Einheiten umzurechnen, indem vordefinierte Tabellenberechnungen oder Berechnungsvorschriften für bestimmte Felder verwendet werden. Eine dynamische Umrechnung im Data-Output ist nicht möglich.	https://help.tableau.com/current/pro/desktop/de-de/calculations_calculatedfields_formulas.htm https://help.tableau.com/current/pro/desktop/de-de/calculations_tablecalculations.htm
DO9	Die Darstellungsform der Abbildungen soll durch den Nutzenden frei gewählt werden können. Gewünschte Darstellungsformen sind: Liniendiagramm, Balkendiagramm, Tortendiagramm, Baumdiagramme, Abweichungen, Korrelationen und Streudiagramme, Häufigkeitsverteilungen, Nominale Vergleiche (wie Blasendiagramme oder Heatmaps), Fluss und Sankey-Diagramme, Netzwerkdiagramme)	o	Eine freie Konfiguration der Darstellungsform ist durch den Nutzenden nicht möglich. Es könnten jedoch verschiedene Visualisierung durch Tableau Stories oder Kombinationsdiagrammen vorbereitet werden.	https://help.tableau.com/current/pro/desktop/de-de/stories.htm https://help.tableau.com/current/pro/desktop/de-de/qs_combo_charts.htm
DO10	Die Visualisierungen in Diagrammen/Abb	+	Die Veröffentlichung von Visualisierungen und Dashboards auf Webseiten kann mittels eingebetteten	https://help.tableau.com/current/pro/desktop/de-de/embed.htm

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	ildungen (Data-Outputs) müssen direkt in Artikel auf der UBA-Webseite (Drupal CMS) eingebunden werden können		Codes oder mit Hilfe der Tableau JavaScript API realisiert werden. Wichtig: Um Ansichten im Internet zugänglich zu machen, wird ein "Gastzugang" benötigt, damit auch nicht eingeloggte Nutzende Inhalte ansehen und explorieren können. Der Gastzugang ist nur in der core-basierten Lizenz von Tableau Server enthalten. Für Drupal 8 existiert ein Tableau Modul, welches ggf. weiterverwendet werden kann.	https://help.tableau.com/current/server/de-de/license_product_keys.htm https://help.tableau.com/current/server/de-de/users_guest.htm https://www.drupal.org/project/tableau_dashboard
DO11	Die Visualisierungen in Diagrammen/Abbildungen (Data-Outputs) soll in der zukünftigen umwelt.info Webseite möglich sein.	+	siehe DO10	
DO12	Die Implementierung soll möglichst CMS-offen (unabhängig von dem bestehenden Drupal System) erfolgen. Es soll geprüft werden ob Inhalte z.B. als reponsive iFrames oder vergleichbares eingebettet werden können.	+	siehe DO10	
DO13	Funktionalitäten die nicht CMS-offen implementiert werden, sind als Drupal-Modul (unter Beachtung der Drupal Code-Konventionen) zu	+	Es gibt ein Drupal Modul "Tableau Dashboard Integration" für Drupal 8. Im November 2021 ist aus der Online-Dokumentation nicht ersichtlich, ob das Modul auch mit Drupal 9 kompatibel ist und den Code-Konventionen entspricht.	https://www.drupal.org/project/tableau_dashboard

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	entwickeln in vollständig lauffähig in das CMS der UBA Webseite zu integrieren.			
DO14	Es muss möglich sein, Dashboards im Drupal CMS einzubinden, die aus mehreren Data-Outputs bestehen	+	Alle Darstellungen in Tableau sind Dashboards. Es wird keine Unterscheidung zwischen Data-Output und Dashboard vorgenommen. Tableau-Dashboards werden aus unterschiedlichen Tableau Ansichten erstellt.	https://help.tableau.com/current/pro/desktop/de-de/dashboards.htm
DO15	Dashboards und Data-Outputs sollen auf verschiedenen Display Größen (Mobilgeräte und größere Bildschirme) gut angezeigt werden.	+	Tableau bietet die Möglichkeit, dass Dashboards direkt für unterschiedliche Layouts hinterlegt werden. Die Anordnung und Größenskalierung können automatisch berücksichtigt werden (Responsive Design).	https://help.tableau.com/current/pro/desktop/de-de/dashboards_dsd_create.htm
DO16	Daten und Abbildungen sollen in unterschiedlichen Formaten zum Download angeboten werden. Daten: CSV, Excel Abbildungen: PNG/JPEG	+	Die Export-Funktion bzw. das Download-Format kann vordefiniert werden. Folgende Formate stehen zur Verfügung: - Bild: .png - Daten: .csv - Kreuztabellen: .csv, .xlsx - PDF - PowerPoint	https://help.tableau.com/current/pro/desktop/de-de/export.htm
DO17	Datensätze die zu einem Thema gehören sollen erkundet werden können. Das heißt, dass die Daten sowie verlinkte Datensätze angezeigt werden.	o	Die Komponente Tableau Catalog des Data Management Add-Ons ermöglicht die Pflege und Freigabe von Metadaten und Verzweigungsinformationen. Laut der Online-Dokumentation ist es jedoch unklar, ob diese "Datendetails" auch für externe Gastnutzende der UBA Webseite veröffentlicht werden können. Internen Mitarbeitenden können diese Explorationsmöglichkeit des Tableau Catalogs im Tableau Server nutzen.	https://help.tableau.com/current/online/de-de/dm_perms_assets.htm#zugreifen-auf-verzweigungsinformationen https://help.tableau.com/current/online/de-de/dm_overview.htm#tableau-catalog https://help.tableau.com/current/online/de-dedm_catalog_overview.htm

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			Zur Integration in Drupal wäre voraussichtlich ein eigenes Modul zu implementieren.	https://help.tableau.com/current/pro/desktop/de-de/data_explore_analyze_interact.htm#verwenden-sie-data-details-um-visualisierungsinformationen-anzuzeigen
DO18	Verknüpfte Datensätze sollen in der Visualisierung mit hinzugefügt werden können, um Daten vergleichen zu können.	-	Es existiert keine Standard-Funktionalität. Verknüpfte Datensätze könnten nur durch vordefinierte Dashboards zeitgleich dargestellt werden. Es existiert keine Standard-Funktionalität zum Verknüpfen von verwandten Datensätzen.	https://help.tableau.com/current/pro/desktop/de-de/environ_workbooksandsheets.htm
DO19	Das Corporate Design ist bei der Gestaltung aller Nutzeroberfläche n zu beachten.	+	Mit der Komponente "Embedded Analytics" können einfache Anpassungen wie Firmenlogs und Farben durchgeführt werden ("white labeling").	https://www.tableau.com/de-de/embedded-analytics
DO20	Es ist zu berücksichtigen wie Data-Outputs auch in die Datensuche des UBA eingebunden werden können.	o	Wenn Data-Outputs als Drupal-Module implementiert werden, kann der Standard Drupal-Mechanismus verwendet werden. Hierzu müssen beim Erstellen der Inhalte in Drupal entsprechende Keywords vergeben werden.	
DE1	Über den Data-Explorer müssen Datensätze des Data Stores auffindbar gemacht werden.	+	Mit Tableau Catalog, Tableau Desktop oder Tableau Prep kann man die Datensätze auffinden und durchsuchen.	
DE2	Metadaten von Datensätzen sollen angezeigt werden können.	+	Die Komponente Tableau Catalog des Data Management Add-Ons ermöglicht die Pflege und Freigabe von Metadaten und Verzweigungsinformationen. Internen Mitarbeitenden können diese Explorationsmöglichkeit des Tableau Catalogs im Tableau Server nutzen. Ebenso existiert	https://help.tableau.com/current/online/de-de/dm_perms_assets.htm#zugreifen-auf-verzweigungsinformationen https://help.tableau.com/current/online/de-de/dm_overview.htm#tableau-catalog

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			für Tableau Server eine Metadaten-API.	https://help.tableau.com/current/online/de-de/dm_catalog_overview.htm https://help.tableau.com/current/pro/desktop/de-de/data_explore_analyze_interact.htm#verwenden-sie-data-details-um-visualisierungsinformationen-anzuzeigen https://help.tableau.com/v2021.1/api/metadata_api/en-us/index.html
DE3	Datensätze sollen als interaktive Diagramme angezeigt werden können.	+	Mit Tableau Desktop können interaktive Visualisierungen (Diagramme, Tabellen) erstellt werden. Mit Tableau Server können Ansichten für weitere UBA Mitarbeitende zur Exploration veröffentlicht werden.	https://www.tableau.com/de-de/products/desktop
DE4	Datensätze sollen tabellarisch angezeigt werden können.	+	siehe DE3	https://www.tableau.com/de-de/products/desktop
DE5	Eine Definition von zusammengehörigen Datensätzen für einen Cube soll möglich sein.	+	Zusammengehörige Datensätze können über die Pflege von Metadaten (Datendetails), insbesondere durch Verzweigungsinformationen, miteinander verknüpft werden. Weiterhin können über gemeinsame Felder Daten aus unterschiedlichen Quellen zueinander in Beziehung gesetzt und kombiniert werden. Zusätzlich zu Verzweigungsinformationen können multidimensionale Datensätze als OLAP Cubes angebunden werden. Die entsprechende Konfiguration muss außerhalb von Tableau stattfinden. Inwieweit diese Informationen für externe Nutzende zur	https://help.tableau.com/current/online/de-de/dm_perms_assets.htm#zugreifen-auf-verzweigungsinformationen https://help.tableau.com/current/online/de-de/dm_perms_assets.htm#addnotes https://help.tableau.com/current/pro/desktop/de-de/relate_tables.htm

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			Exploration verwendet werden können, ist unklar.	
DE6	Es soll möglich sein, alle Cubes aufzulisten.	o	Es ist unklar, ob Verzweigungsinformationen mit z. B. Tableau Catalog gesammelt aufgelistet werden können. Falls OLAP Würfel als Datenquelle genutzt werden, ist eine Auflistung möglich.	
DE7	Es soll möglich sein alle Datensätze aufzulisten, die in einem Cube verwendet werden.	+	Mit Tableau Catalog können Metadaten erfasst werden zu den Tableau-Inhalten (z. B. Visualisierungen), "Externen Assets" (z. B. "Datenbanken und Tabellen, die mit Tableau-Inhalten verknüpft sind") und zu Verzweigungsinformationen, das heißt z. B. zu Beziehungen zwischen den Datenquellen und den Datenvisualisierungen.	https://help.tableau.com/current/online/de-de/dm_perms_assets.htm
DE8	Es soll möglich sein für einen Datensatz anzuzeigen, in welchem Cube er verwendet wird.	+	siehe DE7	
DE9	Für jeden Datensatz sollen fachlich begründete Dimensionen ausgewählt werden können.	+	In Tableau werden die Dimensionen aus der Datenquelle zur Auswahl bereitgestellt.	https://help.tableau.com/current/pro/desktop/de-de/datafields_typesandroles.htm https://help.tableau.com/current/pro/desktop/de-de/data_structure_for_analysis.htm#kategorisieren-von-feldern
DE10	Hierarchische Datensätze sollen miteinander verknüpft werden können.	+	Tableau generiert automatisch Hierarchien für Datumsfelder. Es ist möglich, benutzerdefinierte Hierarchien zu erstellen, um Drilldowns anhand von beispielsweise Feldern wie "Region, Bundesland und Landkreis" zu ermöglichen. Es ist unklar, ob die vom	https://help.tableau.com/current/pro/desktop/de-de/qs_hierarchies.htm

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
			Benutzenden definierten Hierarchien in unterschiedlichen Kontexten verwendet werden können. Falls OLAP-Würfel verwendet werden, sind die Hierarchiestrukturen innerhalb des Würfels zu definieren.	
DE11	Datenauszügen für einzelne Datensätze sollen definiert werden können.	+	Es ist möglich, Bereinigungsschritte oder Filter auf den Datensatz anzuwenden und hierdurch eine reduzierte Ansicht zu erzeugen.	https://help.tableau.com/current/prep/de-de/prep_configure_dataset.htm
DE12	Eine Konfiguration von Datensätzen / Dimensionen für einzelne Data-Outputs soll ermöglicht werden. Es soll konfigurierbar sein, welche weiteren Datensätze zu einem Data-Output potenziell hinzugeladen werden dürfen.	o	Daten können nicht dynamisch hinzugeladen werden. Es ist jedoch möglich, vor der Veröffentlichung gemeinsame Visualisierungen oder Arbeitsmappen für die potenziell zusammen anzuzeigenden Daten vorzukonfigurieren.	
DE13	Konfiguration der initialen und konfigurierbaren Darstellungsform	+	Es ist möglich, eine Standardeinstellung der Felder inklusiver Standardaggregationen und standardmäßigen Sortierreihenfolgen zu definieren.	https://help.tableau.com/current/pro/desktop/de-de/datafields_fieldproperties.htm
DE14	Konfiguration weiterer Optionen des Data-Outputs für die Nutzenden: - Daten hinzuladen - Einheiten umrechnen / Datensatz mit anderen Einheiten laden - Aggregationen berechnen /	o	Dashboards können vielfältig vordefiniert werden. Eine Anpassung durch den Nutzenden ist jedoch nicht mehr möglich.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Datensätze mit anderen Aggregationen laden - Erlaubte Downloads			
DE15	Für jeden Data-Output sollen die Zusatz-Funktionalitäten (z.B. Aggregationen berechnen, Daten hinzuladen, …) bei Bedarf deaktiviert werden können.	o	(De-)Aggregationen können pro Darstellung konfiguriert werden. Hinzuladen von Daten oder Umrechnen von Einheiten ist nicht konfigurierbar.	
DE16	Konfiguration von Dashboards (Kombination aus mehreren Data-Outputs)	+	Dashboards und Stories können erstellt werden.	https://help.tableau.com/current/pro/desktop/de-de/stories.htm https://help.tableau.com/current/pro/desktop/de-de/dashboards.htm
DE17	Der Data-Explorer soll außerhalb des UBA CMS implementiert werden können und hat kein spezifisches CMS zur Voraussetzung.	+	Tableau Desktop und Tableau Server wären die Data-Explorer Komponente. Tableau Desktop ist eine Desktop Anwendung und muss pro Nutzenden installiert werden. Tableau Server ist als Webanwendung verfügbar.	
DE18	Für externe Analysetools soll der Zugriff auf die Daten über eine API ermöglicht werden.	o	Es stehen u. a. eine REST API und eine Metadaten-API bereit. Diese sind jedoch primär zur Steuerung der Tableau Inhalte gedacht. Es ist unklar, ob die API auch zum Abfragen von Daten für externe Nutzende sinnvoll nutzbar ist.	https://help.tableau.com/current/api/rest_api/en-us/REST/rest_api.htm https://help.tableau.com/v2021.1/api/metadata_api/en-us/index.html
DE19	Für externe Analysetools soll der Zugriff auf die Daten über einen Export ermöglicht werden.	+	Tableau Desktop bietet die Möglichkeit des Daten-Exports.	https://help.tableau.com/current/pro/desktop/de-de/save_export_data.htm
DE20	Die Konfiguration der Data-Outputs und Data Cubes	-	Data-Ouputs werden größtenteils in Tableau Desktop konfiguriert. Lediglich die	https://help.tableau.com/current/online/de-de/server_desktop_web

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	soll über ein Webinterface möglich sein.		Bereitstellung erfolgt über eine Webanwendung (Tableau Server).	_edit_differences.htm https://public.tableau.com/views/TableauDesktopvTableauWebEditing/DesktopvsWebEdit?:showVizHome=no
DE21	Es soll möglich sein Zusatzdateien für Berichte (Bilder in verschiedenen Formaten, Daten) herunterzuladen.	+	Folgende Export-Formate stehen zur Verfügung: Bild: .png Daten: .csv Kreuztabellen: .csv, .xlsx PDF PowerPoint	https://help.tableau.com/current/pro/desktop/de-de/export.htm
DE22	Änderungen an den Daten sollen angezeigt werden können (Überarbeitungs modus)	-	Ein expliziter Überarbeitungsmodus wurde nicht gefunden. Für die Rolle "Viewer" kann man sich für Updates und Benachrichtigungen registrieren. Für den Tableau Server Admin gibt es weitere Überwachungsoptionen.	https://help.tableau.com/current/guides/everybody-install/de-de/everybody_admin_monitor.htm
DI1	Es soll eine Nutzeroberfläche und ein Workflow gestaltet werden, um weitere Datenquellen anzubinden.	+	Mit Tableau Prep steht eine Nutzeroberfläche, die diese Anforderungen abdecken kann, zur Verfügung. Dort können „DataFlows“ in einzelnen Schritten definiert werden, um die Verarbeitung, Validierung und Zusammenführung verschiedener Datenquellen zu steuern.	https://help.tableau.com/current/prep/de-de/prep_get_started.html https://help.tableau.com/current/prep/de-de/prep_about.htm#a-tour-of-the-tableau-prep-builder-workspace
DI2	Der Data-Input soll entweder durch die bestehenden Excel Templates der Redaktion oder falls möglich durch eine direkte Anbindung der Datenhaltung der datenhaltenden Stellen durchgeführt werden können.	+	Es sind verschiedene „Connnectoren" vorhanden, um diverse standardisierte und proprietäre Datenbanken und Dokumentformate anzubinden. Der Data-Input ist grundsätzlich jedoch unabhängig von Tableau.	https://help.tableau.com/current/pro/desktop/de-de/exampleconnections_overview.html https://help.tableau.com/current/pro/desktop/de-de/examples_excel.htm

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DI3	Daten sollen beim Import eine Qualitätssicherung durchlaufen	+	Eine automatisierte Qualitätssicherung könnte innerhalb eines Workflows mit Tableau Prep realisiert (bzw. selbst implementiert) werden. Grundsätzlich ist der Daten-Import jedoch unabhängig von Tableau und kann auch mit anderen ETL-Tools realisiert werden.	https://www.tableau.com/de-de/products/prep
DI4	Der Data-Input muss mit verschiedenen Eingangsformaten (Datenbanken, Excel, CSV, WebServices) umgehen können, um Daten von verschiedenen datenhaltenden Stellen einzulesen.	+	Verschiedene Datenquellen sind möglich (Datenbanken, online cloud storages, spreadsheets, …). Eine Liste der möglichen Datenhaltungsquellen ist im Quell-Link zu finden. Siehe auch DI2. Andere Datenquellen müssten durch ein ETL-Tool zunächst aufbereitet werden.	https://www.tableau.com/de-de/products/prep#data-sources
DI5	Metadaten, welche in den Datenquellen enthalten sind, sollen mit ausgelesen werden.	o	Es ist unklar, ob semantische Metadaten der Datensätze automatisiert mit eingelesen werden können. Der Daten-Import kann unabhängig von Tableau durch ein ETL-Tool durchgeführt werden.	https://help.tableau.com/current/online/de-de/dm_perms_assets.htm
DI6	Während des Data-Inputs sollen zusätzliche Metadaten für einen Datensatz angereichert werden können.	+	Die Komponente Tableau Catalog des Data Management Add-Ons und die Metadata API ermöglichen die Pflege von Metadaten. Ergänzend zu DI5 können Metadaten editiert werden.	https://help.tableau.com/current/online/de-de/dm_overview.htm#tableau-catalog https://help.tableau.com/current/api/metadata_api/en-us/index.html
DI7	Während des Data-Inputs sollen Verlinkungen zu anderen Datensätzen im Data-Store möglich sein.	+	In Tableau Prep können Datensätze auf vielfältige Weise zu einem „data flow“ zusammengeführt und gemeinsam verarbeitet werden. Mit der Pflege von Metadaten oder der Metadaten-API können Informationen zu verlinkten Datensätzen bereitgestellt werden.	https://help.tableau.com/current/prep/en-us/prep_build_flow.htm https://help.tableau.com/current/api/metadata_api/en-us/index.html https://www.tableau.com/de-de/products/prep

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DI8	Mit dem Data-Input soll es möglich sein, mehrere Versionen von einem Datensatz einzulesen und im Data-Store zu speichern.	+	Mit der Rolle "Creator" oder "Explorer" ist es möglich mit Inhaltsrevisionen in Tableau Desktop zu arbeiten.	https://help.tableau.com/current/pro/desktop/de-de/qs_revision_history.htm
DI9	Für Datensätze mit gleichbleibenden Strukturen soll ein automatisierter Importprozess möglich sein.	+	Mit Tableau Prep Conductor können Workflows für automatisierte Import-Prozesse definiert werden.	https://www.tableau.com/de-de/products/prep https://help.tableau.com/current/server/de-de/server_process_prep_conductor.htm
DI10	Für Automatisierungen soll es möglich sein, zeitliche Ablaufpläne zu konfigurieren, um Datensätze regelmäßig zu importieren.	+	Mit Tableau Prep Conductor können Workflows für automatisierte Import-Prozesse geplant werden.	https://www.tableau.com/de-de/products/prep https://help.tableau.com/current/server/de-de/server_process_prep_conductor.htm
RP1	Es muss ein redaktioneller Prozess zur Pflege der Daten zur Umwelt entwickelt werden, oder der DataCube muss in den bestehenden Prozess mit eingebettet werden.	+	Der redaktionelle Prozess ist grundsätzlich unabhängig von Tableau. Da jedes in Tableau erstelle Dashboard zunächst privat ist und erst veröffentlich werden muss, können viele Anforderungen an QS und Freigaben einfach umgesetzt werden.	
RP2	Daten dürfen nur nach einer Überprüfung durch einen Data-Output auf der Webseite dargestellt werden.	+	Nur Anwender mit entsprechenden Berechtigungen dürfen Inhalte veröffentlichen. (Rollen "Creator" und "Explorer")	
RP3	Data-Outputs müssen aufwandsarm in Artikel der UBA	+	Verschiedene Einbettungsmöglichkeiten sind möglich (vgl. DO10, DO11)	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Webseite eingebettet werden können.			
RP4	Data-Outputs dürfen nur nach vorheriger Freigabe veröffentlicht werden.	+	Dashboards müssen erst freigegeben werden, bevor diese durch andere Nutzende einsehbar sind.	
RP5	Bei Anpassung der Daten eines Berichts/Artikels sollen Redakteure benachrichtigt werden.	+	Es ist möglich, Anwender für Update-Benachrichtigungen zu registrieren.	
RP6	Der manuelle Aufwand des redaktionellen Prozesses soll soweit möglich reduziert werden.		Unabhängig von Tableau.	
DS1	Es ist eine zentrale Datenhaltung mit allen Daten zur Umwelt aufzubauen.	+	Innerhalb von Tableau wird im Regelfall keine interne Datenhaltung aufgebaut. Es können verschiedene (dezentrale) Datenhaltungsquellen verwendet und ebenso gleichzeitig angebunden werden. Die Verwendung einer relationalen Datenbank kann sinnvoll sein, da diese auch als Quelle für OLAP-Cubes verwendet werden könnte. Falls eine zentrale Datenhaltungskomponente aufgebaut wird, empfiehlt es sich, ein harmonisiertes Datenschema wie z. B. Stern- oder Schneeflockenschema zu verwenden, um den Tableau-Zugriff zu optimieren.	https://help.tableau.com/current/pro/desktop/de-de/cubes.htm https://help.tableau.com/current/pro/desktop/de-de/datasource_datamodel.htm#stern-und-schneeflocke
DS2	Die Datenstrukturen müssen klar definiert werden. Es muss ein themenübergreifender	+	Prinzipiell unabhängig von Tableau. Für eine zentrale Datenhaltung werden Star- oder Snowflake-Schemata empfohlen (vgl. DS1). Basierend auf diesen kann ein	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	Gesamtansatz für das Datenmodell ausgearbeitet werden, in den sich die verschiedenartigen Daten zur Umwelt einordnen.		Gesamtansatz entworfen werden.	
DS3	Das Datenmodell muss so konzipiert werden, dass es einerseits beliebig vertieft/ detailliert und andererseits beliebig fachlich erweitert werden kann.	o	s. DS1 und DS2	
DS4	Die datenhaltenden Stellen sollen ihre Daten möglichst automatisiert in den Data-Store einspielen können.	+	Unabhängig von Tableau. Hierfür kann der Tableau Prep Conductor oder ein anderes ETL-Tool verwendet werden.	
DS5	Auch rückwirkend müssen Datenaktualisierungen und -erfassungen im Data-Store an die Fachsysteme (teil-) automatisiert übergeben werden können.	o	Analog zu DS4. Im Datenmodell muss zusätzlich eine Logik zur Versionsverwaltung abgebildet sein (DI8).	
DS6	Die Webseite soll Daten direkt aus dem Data-Store nutzen.	+	Automatische Updates der (veröffentlichten) Ansichten ist mit Tableau Server möglich.	https://help.tableau.com/current/pro/desktop/de-de/refresh.htm
DS7	Datenänderungen sollen automatisch an die Redakteure bzw. Fachexperten/-innen übermittelt werden.	+	Es ist möglich, Anwender für Update-Benachrichtigungen zu registrieren (vgl. RP5).	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DS8	Die Daten müssen über Status so markiert sein, dass sofort erkennbar ist, in welchem Bearbeitungszust and sie sich gerade befinden.		Unabhängig von Tableau.	
DS9	Dynamische Zugriffe auf die Daten müssen in Abhängigkeit des Status definiert und kontrolliert werden können.	o	Wenn Status-Informationen für alle Datensätze gepflegt werden, können diese in Dashboard Filtern entsprechend verwendet werden.	
DS10	Es muss möglich sein, Daten aus verschiedenen Quellsystemen (insbesondere aus DESTATIS) zu übernehmen, diese einheitlich strukturiert zu verwalten, sodass sie dann als Dimensionen der unterschiedlichen Daten zur Umwelt im Data-Store genutzt werden können.		Unabhängig von Tableau	
DS11	Es muss möglich sein, für alle erfassten Datensätze Dimensionen zu definieren.		Unabhängig von Tableau. Um OLAP-Cubes als Datenquelle und die hiermit verbundenen Dimensionen für Tableau zu nutzen, ist zu beachten, dass einige Tableau-Funktionalitäten für Cube-Datenquellen nicht verfügbar sind. Hierfür existieren jedoch alternative Möglichkeiten, um die fehlenden Funktionalitäten zu kompensieren.	https://help.tableau.com/current/pro/desktop/de-de/cubes.htm
DS12	Es muss möglich sein, die übernommenen Daten (konkrete Werte der Datensätze) den		Unabhängig von Tableau.	

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
	entsprechenden Dimensionen zuzuordnen.			
DS13	Die Werte müssen auch als Werte (Zahlen, Vektoren) gespeichert werden, sodass mit ihnen gerechnet werden kann.		Unabhängig von Tableau.	
DS14	Es soll möglich sein, im Data-Store Berechnungen zu hinterlegen, die automatisch ausgeführt werden.	o	Unabhängig von Tableau. Um OLAP-Cubes als Datenquelle und die hiermit verbundenen "Vorberechnungen" für Aggregationen in Tableau zu nutzen, ist zu beachten, dass einige Tableau-Funktionalitäten für Cube-Datenquellen nicht verfügbar sind. Hierfür existieren jedoch alternative Möglichkeiten, um die fehlenden Funktionalitäten zu kompensieren. Berechnungen könnten grundsätzlich auch im Data-Input durchgeführt werden.	https://help.tableau.com/current/pro/desktop/de-de/cubes.htm
DS15	Es soll möglich sein, bestimmte Berechnungen manuell anzustoßen.	+	In Tableau Desktop können Berechnungen beim Erstellen von Dashboards durchgeführt werden. Alternativ könnten Berechnungen unabhängig von Tableau im Data-Input durchgeführt werden (ETL-Tool, z.B. Tableau Prep).	
DS16	Es soll möglich sein, Berechnungen hierarchisch auszuführen. Das gilt auch für automatisierte Berechnungen.		Unabhängig von Tableau. Um OLAP-Cubes als Datenquelle und die hiermit verbundenen "Hierarchien" in Tableau zu nutzen, ist zu beachten, dass einige Tableau-Funktionalitäten für Cube-Datenquellen nicht verfügbar sind. Hierfür existieren jedoch alternative Möglichkeiten, um die fehlenden Funktionalitäten zu kompensieren.	https://help.tableau.com/current/pro/desktop/de-de/cubes.htm

Num-mer	Beschreibung	Bewer-tung	Umsetzung	Quelle
DS17	Es muss auch weiterhin möglich sein, extern mit den Daten zur Umwelt zu arbeiten und externe Daten mit ihren Bezügen zu den Ausgangsdaten im Data-Store einzuspielen.	-	Unabhängig von Tableau. Es müsste jedoch ein Export/Import Workflow zur Bearbeitung von Daten außerhalb des Tools etabliert werden. Dies könnte z.B. durch ETL-Prozesse realisiert werden.	
DS18	Die Werte müssen historisch verwaltet werden.	o	Unabhängig von Tableau. Es muss jedoch ein Historisierungs-Ansatz gefunden werden, der mit dem von Tableau empfohlenen Datenmodell funktioniert.	
DS19	Bei hierarchischen Berechnungen müssen die Bezüge der Daten jederzeit wieder selektierbar sein.		s. DS18	
DS20	Objektklassen, die für die Dimensionierung herangezogen werden, müssen ebenfalls historisch verwaltet werden, um ältere Daten zur Umwelt auch weiterhin einordnen zu können.		s. DS18	

3.5.5.3 Bewertung

Tableau ist ein etabliertes Tool für Business Intelligence (Abkürzung: BI) Prozesse und stellt vielfältige Gestaltungsmöglichkeiten für Visualisierungen und Dashboards bereit. Einerseits zeigt dies eine gewisse Marktreife der Lösung. Andererseits wird durch die verschiedenste Visualisierungsoptionen eine hohe Abdeckung der komplexen Anforderungen an die Data-Output Komponente und den Data-Explorer erzielt. Während für interne Nutzende über die Tableau Desktop und Tableau Server Komponente eine umfassende Datenexploration gewährleistet wird, bleibt anzumerken, dass es gewisse Restriktionen für externe Nutzende

geben würde. In diesem Kontext ist zu sehen, dass das Hinzuladen von Datensätzen für interne Mitarbeitende über die Tableau Anwendungen möglich ist. Externen Nutzende werden vorgefertigte Dashboards und Ansichten über die Webseite bereitgestellt. In diesem Fall ist es derzeit nicht abschließend festzustellen, ob es auch möglich wäre als externer Nutzender Datensätze hinzuzuladen. Es könnte sein, dass bei der Erstellung der Ansicht oder des Dashboards intern bereits Datensätze vorab zum Hinzuladen hinterlegt werden könnten. Auf Basis der Online-Dokumentation ist dies jedoch nicht endgültig zu bewerten. Gleiches betrifft die ad-hoc Auswahl eines anderen Diagrammtyps durch externe Benutzende der UBA Webseite: Während der Wechsel zwischen vordefinierten Diagrammtypen in einer Ansicht denkbar wäre, wird die Auswahl eines beliebigen Diagrammtyps als schwierig gesehen. Insgesamt wird es so bewertet, dass durch die Redaktion die verschiedenen Visualisierungsmöglichkeiten in den Tableau Ansichten und Dashboards hinterlegt werden sollten und diese dann zur Nutzung auf der Webseite freigegeben werden könnten. In diesem Sinne ist eine gewisse Antizipation der verschiedenen Explorationsoptionen durch interne Mitarbeitende notwendig und eine komplett freie und unbegrenzte Datenerkundung durch externe Nutzende kritisch und somit letztlich als beschränkt einzuordnen.

Eine weitere Frage, die sich im Zusammenhang mit dem Gastzugriff nicht auf Basis der Online-Recherche final bewerten lässt, ist die bezüglich anfallender Lizenzkosten. Während für die „rollenbasierte Lizenzmetrik“ (Tableau Software, LLC, o. J. - h) Lizenzkosten auf der Webseite veröffentlicht sind (vgl. (Tableau Software, LLC, o. J. - i)), sind für das „kernbasierte Lizenzmodell“ (Tableau Software, LLC, o. J. - h) keine Preisinformationen veröffentlicht. Im Kontext des Data Cube Projektes ist die kernbasierte Lizensierung notwendig, da nur diese die „Gastbenutzer-Option“ unterstützt (Tableau Software, LLC, o. J. - h) (Tableau Software, LLC, o. J. - f). Darüber hinaus ist eine Einordnung der anfallenden Kosten derzeit nicht möglich, da weiterhin keine Preisinformationen für „Eingebettete Analytics“ online veröffentlicht sind (Tableau Software, LLC, o. J. - g). Diese Zusatzkomponente sollte jedoch für das Data Cube Projekt ebenfalls verwendet werden. Enthaltene Schnittstellen sind zwingend erforderlich, um die Data Cube Anforderungen bestmöglich abzudecken. Beispielsweise wird durch die JavaScript API eine Veröffentlichung von Ansichten und Dashboards in Webseiten vereinfacht. Zudem ist mit der Zusatzkomponente auch eine gewissen Anpassbarkeit gegeben, um das UBA Design in Tableau und erstellten Ansichten zu berücksichtigen.

Die Integration von Tableau Ansichten im Drupal CMS des UBA wird als besondere Herausforderung betrachtet. Während für Drupal 8 ein Tableau Modul für Dashboard Integrationen existiert (Avbar, Norton, & Chadwick, o. J.), ist derzeit unklar, ob für Drupal 9 ebenfalls zur Zeit der Umsetzung ein solches Modul bereitstehen würde. Zum aktuellen Zeitpunkt ist die Online-Recherche in diese Richtung nicht erfolgreich. Da der Source-Code für das referenzierte Modul frei verfügbar ist (Drupal, o. J.) und unter der GNU General Public Licence, Version 2 (Free Software Foundation (FSF), 1991 - a) steht, ist es denkbar, dass der Programmiercode für die Entwicklung eines Drupal 9 Moduls hilfreich sein könnte. Bevor eine solche Eigenentwicklung angestrebt wird, sollten jedoch tiefergehend die Integrationsmöglichkeiten in die Webseite durch Tableau Funktionalitäten analysiert werden. Hierzu wird von Tableau Einbettungscode bereitgestellt und zusätzlich sind Tableau JavaScript Schnittstellen für Webentwicklungen vorhanden (Tableau Software, LLC, o. J. - c).

Durch eine hohe Benutzerfreundlichkeit und die weite Bandbreite an Darstellungs- und Explorationsoptionen erscheint es möglich, dass zunächst intern mit geringem Zeitaufwand erste Visualisierungen erstellt werden könnten, die dann extern auf der Webseite veröffentlicht werden könnten.

3.5.6 Vergleich der Lösungskomponenten

Durch die Darstellung der verschiedenen Lösungskomponenten in den vorherigen Kapiteln hat sich herauskristallisiert, dass keine der genannten Optionen alle Anforderungen abdecken könnte. Vielmehr existieren für jede Lösungsalternative Limitierungen, denen durch Eigenentwicklung oder die Kombination mit anderen Lösungsprodukten entgegengewirkt werden könnte. Die Lösung auf Basis von Highcharts nimmt in diesem Kontext eine Sonderrolle ein, da durch den massiven Anteil an Eigenentwicklung ein hohes Maß an Flexibilität gewährt wird und somit eine hohe Anforderungsabdeckung möglich wäre. Im Folgenden sollen die verschiedenen Lösungskomponenten zusammenfassend verglichen werden. Dabei wird vor allem auf die grundlegenden Unterschiede sowie mögliche Kombinationen der einzelnen Komponenten eingegangen. Detailbeschreibungen der einzelnen Lösungskomponenten hingegen sind den vorherigen Kapiteln zu entnehmen.

Tabelle 7: Zusammenfassende Bewertung der einzelnen Lösungskomponenten

Anforderung	.Stat Suite	Tableau	Sisense	Mesap	Highcharts
Abdeckung fachliche Anforderungen	+	+	+	o	+
Gute Anpassbarkeit	o	-	-	-	+
Kosten Eigenentwicklung	€€	€	€	€€	€€€
Lizenz-Kosten	€	€€€	€€€	€€	€
Marktreife	o	+	+	o	-
Geringer Aufwand Installation	-	+	+	+	o
Geringer Aufwand Betrieb / Updates	-	+	+	+	o

Zunächst werden die Lösungskomponenten .Stat Suite, Tableau und Sisense verglichen, da diese als fertige Lösungen betrachtet werden können. Mesap und Highcharts sind etwas differenzierter zu betrachten und folgen anschließend.

Als Gemeinsamkeit über die vorgestellten Lösungskomponenten hinweg ist die fehlende Integration in Drupal zu erwähnen. Sowohl Tableau als auch Sisense bieten die Möglichkeiten, Dashboards zu erzeugen, welche über eine Integration API auch ohne iFrames in Webseiten eingebettet werden können. Diese Integration muss jedoch als eigenes Drupal-Modul neu implementiert werden. Für Tableau existiert bereits eine open-source Entwicklung für Drupal 8, die möglicherweise verwendet werden kann. Für die .Stat Suite ist keine direkte Data-Output Möglichkeit vorhanden. Es ist grundsätzlich möglich, die im .Stat Data-Explorer erzeugten Diagramme zu teilen, allerdings können hierbei nur iFrames verwendet werden, welche im Data Cube Projekt nicht verwendet werden können (siehe Kapitel 2.6). Da sich die geteilten Data-Outputs jedoch auf reine Diagramme ohne jegliche Interaktivität (Filterungen, etc.) beziehen, müsste hier voraussichtlich nicht nur die Drupal-Integration, sondern auch die gesamte Interaktivität implementiert werden. Es ist denkbar, den Programmcode des .Stat Data-Explorer an dieser Stelle weiterzuverwenden.

Das Explorieren von Datensätzen durch externe Nutzende ist als Standardfunktionalität nur durch den .Stat Data-Explorer abgedeckt. Dort können Datensätze gesucht, gefiltert und dargestellt werden. Dimensionen des jeweiligen Datensatzes können dabei beliebig aktiviert/deaktiviert werden. Eine Verlinkung von verschiedenen Datensätzen ist nicht möglich. Tableau und Sisense bieten keine Standard-Funktionen zur Exploration. Für beide Lösungen müssten Dashboards mit entsprechenden Möglichkeiten vordefiniert werden. Es ist z. B.

denkbar, mehrere Datensätze für Vergleiche auf einem Dashboard zu visualisieren und diese gegebenenfalls erst durch Nutzerinteraktionen sichtbar zu schalten. Ein freies Suchen von Datensätzen ist jedoch nicht möglich.

Sisense und Tableau sind zur Verbindung mit existierenden Datensätzen konzipiert. Beide Produkte können sich mit einer Vielzahl von Datenquellen (Dateien und Datenbanken) verbinden und existierende Strukturen verwenden. Dabei ist es in beiden Lösungen möglich, Daten in einer internen, proprietären Datenbank (Tableau: Hyper, Sisense: ElastiCube) neu zu strukturieren und für verschiedene Anwendungsfälle aufzubereiten. Dadurch werden unter anderem Cube-Funktionalitäten ermöglicht. Die Data-Store und Data-Input Komponente kann daher für beide als unabhängig vom jeweiligen Produkt betrachtet werden. Tableau definiert jedoch verschiedene Vorgaben für Datenbankmodelle (Stern- oder Schneeflockenschema), um Deaggregationen zu ermöglichen. Darüber hinaus ist die Anbindung von OLAP-Cubes in Tableau grundsätzlich möglich, doch können dadurch bestimmte Funktionalitäten nicht genutzt werden und müssten gegebenenfalls durch Eigenentwicklung ergänzt werden. Durch die Sisense Dokumentation waren keine Vorgaben an das Datenbankmodell auffindbar. Im direkten Vergleich dazu hat die .Stat Suite strikte Vorgaben an die Datenhaltung. Alle Daten müssen über das SDMX Format importiert werden. Die interne Datenbank soll dabei nicht direkt, sondern nur über die Schnittstellen befüllt werden. Durch SDMX müssen für alle neuen Datensätze zunächst die Strukturen und Dimensionen beschrieben werden, anschließend können Daten importiert werden. Es können daher nur Informationen in die Datenhaltung übernommen werden, die durch SDMX abgebildet werden können. Die Erstellung der SDMX Dateien muss dabei durch externe Tools (z. B. ETL-Tools) durchgeführt werden. Der Data-Store ist damit vorgegeben. Für den Data-Input können die Upload-Webseiten des .Stat DLM verwendet werden, nachdem die SDMX Daten erzeugt wurden. Dieser Ansatz erscheint besonders sinnvoll, wenn die Datenlieferungen bereits in SDMX erfolgen würden, was jedoch im UBA derzeit nicht abzusehen ist. Es ist an dieser Stelle mit einem erheblichen Aufwand und Wissensaufbau zu rechnen, um Daten erfolgreich über die SDMX Schnittstellen zu bearbeiten.

Sisense und Tableau erscheinen auf den ersten Blick ähnlich von den angebotenen Funktionalitäten zur Erstellung von Dashboards. Es gibt jedoch ein paar grundlegende Unterschiede. Tableau besteht aus mehreren Komponenten (z. B. Tableau Desktop, Tableau Server, Tableau Prep), die im Zusammenspiel sinnvoll für den Data Cube erscheinen. Dabei können Dashboards in einer Desktop-Anwendung erzeugt werden. Sisense ist als einzelne Webanwendung konzipiert. Alle Konfigurationen können im Browser durchgeführt werden. Sowohl Tableau als auch Sisense sind für Embedded-Analytics ausgelegt, wodurch eine Integration in andere Anwendungen möglich ist. Tableau könnte jedoch auch nur mit dem Tableau Desktop für interne Nutzungen betrieben werden. Im Vergleich zu Sisense bietet Tableau über die Basis-Funktionalitäten hinaus auch noch weitere Werkzeuge für den Data-Input (Tableau Prep). Für Sisense sind keine Data-Input Tools vorhanden. Für beide Produkte sind keine ausreichenden Kosten auf der Webseite dargestellt, weshalb eine Preisanfrage durch das UBA für eine weitere Unterscheidung notwendig ist.

Die beste API wird von den drei Lösungskomponenten durch die .Stat Suite bereitgestellt. Mit der SDMX-API können Daten und Strukturen abgefragt und für andere Anwendungen nutzbar gemacht werden. Die API-Endpunkte werden dabei im Data-Explorer direkt zur Nutzung angezeigt. Tableau und Sisense bieten beide auch eine Vielzahl von APIs. Diese sind jedoch meistens zur Steuerung der Dashboards und der internen Orchestrierung gedacht. Der Zugriff für konkrete Daten ist komplizierter und für Nutzende der Dashboards nicht direkt ersichtlich.

Da es sich bei Tableau und Sisense um etablierte Produkte handelt, wird bei beiden Lösungen davon ausgegangen, dass Installation, Betrieb sowie die initiale Nutzung mit vergleichsweise

geringem Aufwand ermöglicht werden können. Dazu steht für beide Produkte ein deutschsprachiger Support bereit, und es werden Schulungen angeboten. Bei der .Stat Suite hingegen handelt es sich um ein Community Projekt, welches aus vielen verschiedenen Komponenten besteht. Es ist von einem erheblichen Aufwand bei der Erstinstallation und auch bei möglichen Software-Updates auszugehen. Hinsichtlich Support ist anzumerken, dass für .Stat Suite nur für zahlende Mitglieder eine technische Unterstützung verfügbar ist. Eine öffentliche Community durch Foren oder ähnliches scheint nicht zu existieren.

Nachdem die kommerziellen Lösungen Tableau und Sisense mit der open-source Alternative .Stat Suite verglichen wurden, werden nachstehend die Ansätze basierend auf Mesap und Highcharts abschließend erörtert. Mesap ist als Lösungskomponente nur für den Data-Store und den Data-Explorer zu verwenden. Grundsätzlich könnte das Produkt zur Datenhaltung des Data Cube verwendet werden. Durch das absehbare Ende der Unterstützung durch den Hersteller ist die derzeit im UBA verwendete Version von Mesap jedoch nicht sinnvoll für neue Projekte zu verwenden. Hier sollte zunächst die neue Version „Seven2one TechStack“ evaluiert werden, zu der aktuell noch keine Informationen vorliegen. Ein großer Vorteil von Mesap wären Synergieeffekte durch die breite Verwendung im UBA gewesen. Die neue Version der Software ist jedoch noch nicht bekannt. Zum derzeitigen Stand ist auch noch keine langfristige Strategie bezüglich des Umgangs innerhalb des UBA mit dem auslaufenden Produkt und den damit verbundenen Wartungsaufwänden und Updateszenarien abschließend geklärt. Im Folgenden soll jedoch grob auf mögliche Kombinationen eingegangen werden. Da Tableau und Sisense auf existierende Datenhaltungen aufsetzen, wäre zu evaluieren, ob diese über eine API an das Nachfolgeprodukt von Mesap, den TechStack, angebunden werden können. Dabei wäre zu überprüfen, ob alle Informationen durch die BI-Tools weiterhin nutzbar sind. Vor allem die Möglichkeiten zur Aggregation/Deaggregation müssten sich sinnvoll ergänzen, damit eine Kombination der Lösungen sinnvoll wäre.

Der größte Anteil an Eigenentwicklungen ist durch Highcharts zu erwarten. Highcharts bietet lediglich eine JavaScript Bibliothek zur Umsetzung eigener interaktiver Diagramme. Dieser Ansatz ermöglicht es jedoch, langfristig alle Anforderungen des UBA an den Data Cube zu implementieren. Im Vergleich zu Sisense und Tableau sind dabei fast keine Lizenz-Kosten zu berücksichtigen. Da Highcharts jedoch nur die grundlegenden Chart-Funktionen bietet, müssten alle anderen Komponenten (Oberflächen für Data-Explorer/Data-Output, APIs, Nutzermanagement) neu implementiert werden. Für Data-Input und Data-Store könnten andere existierende Lösungskomponenten verwendet werden. Für die Data-Store Komponente könnte an dieser Stelle möglicherweise auf den Seven2one TechStack zurückgegriffen werden. Eine vollständige Abdeckung aller Anforderungen während der geplanten Projektlaufzeit ist als kritisch zu bewerten. Durch eine Sprint-basierte Umsetzungsphase und die damit einhergehende starke Priorisierung der Anforderungen erscheint es dennoch möglich, eine solide Basis für zukünftige Weiterentwicklungen und mögliche Anschlussprojekte zu realisieren.

Zusammenfassend sind vor allem die drei Ansätze .Stat Suite, Eigenentwicklung (ggf. mit Highcharts und Mesap) oder existierende BI-Tools denkbar. Sisense und Tableau sind sich sehr ähnlich, weshalb eine Entscheidung zwischen diesen Komponenten möglicherweise durch unterschiedliche Preise getroffen werden kann. Ein großer Nachteil der BI-Tools ist die eingeschränkte Exploration über die Webseite. Die .Stat Suite bietet eine gute Exploration, aber keine integrierbaren Data-Outputs und Dashboards. Darüber hinaus ist der Betrieb voraussichtlich schwierig und das Datenformat SDMX ein zusätzlicher Komplexitätsfaktor. Eine Eigenentwicklung bietet viele Gestaltungsfreiräume zur Abdeckung aller Anforderungen, verbunden jedoch mit einem großen Entwicklungsaufwand.

4 Lösungsansatz basierend auf .Stat Suite

Basierend auf den Darstellungen der verschiedenen Lösungskomponenten aus Kapitel 3.5 wurde die .Stat Suite der Statistical Information System Collaboration Community (SIS-CC) für die Umsetzung des Data Cube Projektes durch das UBA ausgewählt. Ausschlaggebend für diese Entscheidung waren insbesondere die große Abdeckung der Anforderungen, die Flexibilität der Lösung sowie die Open Source Strategie der SIS-CC. Des Weiteren verfügt die .Stat Suite über eine sehr aktive Community, die einen guten Indikator für Zukunftssicherheit und aktive Weiterentwicklung darstellt. Mit der OECD existiert außerdem eine wichtige Referenz-Organisation, die die .Stat Suite auf vergleichbare Weise nutzt. Im folgenden Kapitel soll daher der Lösungsansatz detaillierter dargestellt werden.

Da die .Stat Suite für das Austauschformat SDMX entwickelt wurde und dieses Format auch für den Data-Import notwendig ist, werden zunächst grundlegende Begriffe und Konzepte des Formats beschrieben. Anschließend wird der redaktionelle Prozess auf Basis der Lösungskomponente erläutert.

4.1 Einführung in SDMX

Zur Einführung in das Thema SDMX sollen zunächst verschiedene SDMX-Begriffe im Kontext des Data Cube Projektes beschrieben werden. Anschließend werden die wichtigsten Klassen und deren Nutzung erläutert.

4.1.1 Das Vokabular des SDMX für den Lösungsansatz

In der folgenden Tabelle werden einige der in der Konzeptionsphase entwickelten Begriffe (s. Kapitel 3.1) wieder aufgegriffen. Rechts werden Klassen des SDMX-Datenmodells aufgeführt, die sich für die Umsetzung am besten eignen. Im Anschluss werden diese und weitere Klassen kurz dargestellt.

Tabelle 8: Zusammenhang zwischen Konzepten im Data Cube und SDMX-Begrifflichkeiten

Zentrale Begriffe im Data Cube Kontext	Erläuterung / Definition	Geeignete Klassen (SDMX 2.1) für die Umsetzung	Anmerkung zur Umsetzung
Datensammlung	Zusammenstellung von Werten zu einem Thema (z.B. eine Excel-Tabelle)	DataSet	
Beobachtung (z. B. eine Zelle in einer Tabelle)	„Werte beschreiben die konkreten Inhalte.“ (s. Kapitel 3.1.1.2)	Observation / ObservationValue	
Dimension	„Dimensionen beschreiben Kategorien oder Interessensgebiete, in denen Daten vorliegen.“ (s. Kapitel 3.1.1.2)	Dimension / TimeDimension	Die Beschreibung der Kategorien / Interessensgebiete erfolgt in gesonderten Elementen (u.a. CodeList, Concept…), auf welche die Dimensionen referenzieren.

Zentrale Begriffe im Data Cube Kontext	Erläuterung / Definition	Geeignete Klassen (SDMX 2.1) für die Umsetzung	Anmerkung zur Umsetzung
Metadaten (semantisch)	„Semantische Metadaten […] beinhalten Informationen über den Inhalt, räumlich-zeitliche Bezüge, die Datenqualität, Zugangsmöglichkeiten oder Nutzungsrechte und beschreiben damit die Eignung von Daten für bestimmte Anwendungszwecke, Präsentations- und Verarbeitungsmethoden‘“ (s. Kapitel 3.1.1.3)	DataFlow	Der DataFlow definiert selbst kaum Metadaten, sondern verbindet andere Strukturelemente (z.B. DataStructures und DataSets...). Daher definiert der DataFlow für einen konkreten Kontext, welche Metadatenelemente relevant sind.
Metadaten (technisch) / Ontologien	„Sie beschreiben die Struktur von Datensammlungen und Regeln der Datenverarbeitung, z. B. Definitionen der Objekte, ihrer Verknüpfungen, der einzelnen Datenfelder und ihrer Verarbeitungs-konventionen“ (s. Kapitel 3.1.1.3)	DataStructure	
Cube	„Zur Veranschaulichung werden Daten mit mehreren Dimensionen häufig als Würfel (Cubes) beschrieben. Dabei sind die Dimensionen die einzelnen Achsen des Würfels.“ (s. Kapitel 3.1.1.2)	Ein konkreter, erfolgreich in die .Stat Suite eingepflegtes DataSet.	Das DataSet wird beschrieben durch eine DataStructure und einen DataFlow.

4.1.2 Die wichtigsten Klassen des SDMX-Modells

Für den hier relevanten Standard SDMX 2.1 liegt ein UML-Modell vor, welches aus zahlreichen Klassen besteht (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c) (S. 3-4). An dieser Stelle werden nur die wichtigsten und für den Anwendungsfall relevanten Klassen vorgestellt. Innerhalb von SDMX wird strikt zwischen Strukturdefinitionen (die Beschreibung der Datenstruktur) und den eigentlichen (Fach-)Daten unterschieden, welche im folgenden Abschnitt vorgestellt werden. Die Schreibweise der Klassennamen entspricht den Formulierungen in den SDMX-Standarddokumenten.

4.1.2.1 Daten

Mit Daten sind an dieser Stelle die konkreten (Fach-)Daten gemeint. Im Kontext des Data Cube sind das üblicherweise Messwerte, oder berechnete Werte. Sie machen den Inhalt der Informationen aus und verweisen auf die sie beschreibenden Strukturdefinitionen. Neben der im Bild dargestellten Formatierung der Daten als XML-Dokument, kann auch SDMX-CSV genutzt werden (SDMX Technical Standards Working Group - SDMX TWG, 2021).

DataSet

Eine Ansammlung von Daten, die einer bestimmten Gliederung folgt, wird als DataSet beschrieben (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c) (S. 68). Ein DataSet enthält selbst wenige Attribute und ist hauptsächlich relevant als eine Art Hülle um eine Menge an Observations (siehe Abbildung 10).

Observation

Eine Observation beschreibt das kleinste Datenelement, in der Regel eine Zelle in einer Tabelle, welche konkret den Wert eines beobachteten Phänomens beschreibt. Die Observation steht dabei im Bezug zu ihren Dimensionen (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)(S. 72).

Es besteht aus

- einem „ObsValue“: der Wert selbst, in Abbildung 10 z. B. „749.212“,
- beliebig vielen „ObsKey“: Mittels der ObsKey wird der Bezug von ObsValues zu den Dimensionen hergestellt, dem der aktuelle Wert zugeordnet wird. Im Beispiel wird der Wert der Zeitperiode (TIME_PERIOD) 2015 und der CO2-Emmission „energy“ zugeordnet.

Abbildung 10: SDMX Strukturelement für Observationen

```
<DataSet action="Information" structureRef="uba_structure_id" >

 <generic:Obs>
   <generic:ObsKey>
     <generic:Value id="TIME_PERIOD" value="2015" />
     <generic:Value id="CO2_Emission" value="energy" />
   </generic:ObsKey>
   <generic:ObsValue value="749.212"/>
 </generic:Obs>
 <generic:Obs>
   <generic:ObsKey>
     <generic:Value id="TIME_PERIOD" value="2015" />
     <generic:Value id="CO2_Emission" value="fuel_combustion" />
   </generic:ObsKey>
   <generic:ObsValue value="746.838"/>
 </generic:Obs>
 <generic:Obs>
   <generic:ObsKey>
     <generic:Value id="TIME_PERIOD" value="2015" />
     <generic:Value id="CO2_Emission" value="energy_industries" />
   </generic:ObsKey>
```

Quelle: eigene Darstellung, con terra GmbH

Attribute

Durch Attribute können verschiedenen SDMX-Elementen weitere Eigenschaften zugeschrieben werden. Diese können, wie in Abbildung 11, auf eine Codelist verweisen oder direkt als Text eingefügt sein (vgl. (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c), S. 73). Attribute können sowohl für Observations als auch für ganze DataSets eingefügt werden.

Abbildung 11: Attribute eines DataSet und einer Observation

```
<DataSet action="Information" structureRef="uba_structure_id" >
  <generic:Attributes>
    <generic:Value id="DATA_SOURCE" value="Emissionswerte"/>
  </generic:Attributes>

  <generic:Obs>
    <generic:ObsKey>
      <generic:Value id="TIME_PERIOD" value="2015" />
      <generic:Value id="CO2_Emission" value="energy" />
    </generic:ObsKey>
    <generic:ObsValue value="749.212"/>
  <generic:Attributes>
    <generic:Value id="UNIT_MEASSURE" value="KG"/>
    <generic:Value id="UNIT_MULT" value="1000"/>
    <generic:Value id="COMMENT" value="Gerundet auf drei Dezimastellen"/>
  </generic:Attributes>
  </generic:Obs>
```

Quelle: eigene Darstellung, hrd.consulting

4.1.2.2 Strukturdefinitionen

Strukturdefinitionen werden in SDMX zur Definition der technischen Metadaten verwendet. Sie sind in XML formatiert und können anschließend durch die Fachdaten referenziert werden.

DataFlow

Ein DataFlow beschreibt das abstrakte Konzept vom Fluss von Daten, die (interne oder externe) Datenlieferanten zu bestimmten Referenzzeitpunkten bereitstellen können (vgl. S.61 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)). Dabei liegt der Fokus auf der Struktur und nicht auf konkreten Ausprägungen der Daten. Ein DataFlow definiert und verbindet die Elemente des Datenaustauschs:

- Er kann einem DataSet (vgl. S. 55/56 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)) zugeordnet werden, dessen Kontext er definiert.
- Er kann einer DataStructure zugeordnet werden, welches die Struktur dieses DataSets definiert (vgl. S.75 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)).
- Er kann beteiligte Organisationen referenzieren (Von wem kommen welche Daten/Metadaten? Wer hat sie erhoben/definiert? An wen sollen sie geliefert werden? Welchem Zweck dient der Datenaustausch?) (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)(S. 73).

DataStructure

Die DataStructure beinhaltet Metadaten und Strukturinformationen, die auch für die Sammlung oder die Verteilung der Daten verwendet werden (vgl. S.61 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)). DataStructures werden durch DataFlows referenziert

und sind somit die strukturelle Grundlage für alle zu beschreibenden Daten. Dadurch können DataStructure verwendet werden, um DataSets zu validieren (vgl. S.75 und 85 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)), was unter anderem auch beim Import von Daten in die .Stat Suite erfolgt. Die Komponente .Stat Suite Data Lifecycle Manager akzeptiert nur DataSets, die erfolgreich validiert wurden. Daher können grundsätzlich nur DataSets importiert werden, deren DataStructure bereits zuvor importiert wurde. In der folgenden Abbildung 12 wird exemplarisch eine DataStructure vorgestellt. In ihr wird definiert, welche Dimensionen ("CO2_Emission" und "TIME_PERIOD") und welche Werte zugeordnete Daten annehmen dürfen. Zuletzt sei angemerkt, dass jeder DataFlow exakt eine DataStructure referenzieren muss.

Abbildung 12: Beispiel einer SDMX Strukturdefinition bestehend aus DataStructureComponents

```
<structure:DataStructure id="DSD_UBA" agencyID="UBA" version="1.0" isFinal="false">
  <common:Name xml:lang="en">DSD for UBA</common:Name>
  <structure:DataStructureComponents>
    <structure:DimensionList id="DimensionDescriptor">

      <structure:Dimension id="CO2_Emission" position="1">
        <structure:ConceptIdentity>
          <Ref id="CO2_Emission" maintainableParentID="CS_COMMON" maintainableParentVersion="1.0"
               agencyID="UBA" package="conceptscheme" class="Concept" />
        </structure:ConceptIdentity>
        <structure:LocalRepresentation>
          <structure:Enumeration>
            <Ref id="CL_CO2_Emission" version="1.0" agencyID="UBA" package="codelist"
                 class="Codelist" />
          </structure:Enumeration>
        </structure:LocalRepresentation>
      </structure:Dimension>

      <structure:TimeDimension id="TIME_PERIOD" position="2">
        <structure:ConceptIdentity>
          <Ref id="TIME_PERIOD" maintainableParentID="CS_COMMON" maintainableParentVersion="1.0"
               agencyID="UBA" package="conceptscheme" class="Concept" />
        </structure:ConceptIdentity>
        <structure:LocalRepresentation>
          <structure:TextFormat textType="ObservationalTimePeriod" />
        </structure:LocalRepresentation>
      </structure:TimeDimension>

    </structure:DimensionList>

    <structure:MeasureList id="MeasureDescriptor">
      <structure:PrimaryMeasure id="OBS_VALUE">
        <structure:ConceptIdentity>
          <Ref id="OBS_VALUE" maintainableParentID="CS_COMMON" maintainableParentVersion="1.0"
               agencyID="UBA" package="conceptscheme" class="Concept" />
        </structure:ConceptIdentity>
        <structure:LocalRepresentation>
          <structure:TextFormat textType="Double" />
        </structure:LocalRepresentation>
      </structure:PrimaryMeasure>
    </structure:MeasureList>
  </structure:DataStructureComponents>
</structure:DataStructure>
```

Quelle: eigene Darstellung, con terra GmbH

Dimension / TimeDimension

Die Dimension und die TimeDimension sind Concepts, um statistische Daten und Zeitreihen zu klassifizieren und zu gruppieren. Dimensions beschreiben die Dimensionen einer DataStructure. Sie können auf Concepts, Codelists und weitere Entitäten verweisen. Im vorliegenden Beispiel (vgl. Abbildung 12) verweist die Dimension ”CO2_Emission” auf das gleichnamige Concept sowie die Codelist ”CL_CO2_Emission". Pro DataStructure (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)darf maximal eine TimeDimension definiert werden. Sie ist eine spezielle Konstruktion, um eine zeitliche Dimension abzubilden und unterliegt einigen Einschränkungen und Besonderheiten (vgl. S.57ff. in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)). Für jede Dimension muss eine LocalRepresentation definiert werden. Darin wird das Format der Werte definiert. Diese können einfache Datentypen (z.B. Typ "Double") oder komplexe Strukturen sein (z.B. "ObservationalTimePeriod" als Format der TimeDimension in Abbildung 12). Für Dimensionen sind in SDMX Enumerations oder Facets vorgesehen. Enumerations beinhalten abgeschlossene Wertelisten (Codelists), welche definieren, welche Einträge in der Dimension erlaubt sind. Facets beschreiben nicht abgeschlossene Werte wie Zahlen oder Texte, welche durch ihre Datentypen (Double, Text, Date, etc.) angegeben werden. In der .Stat Suite sind jedoch nur enumerated Codelists und keine Facets als Dimensions erlaubt.

PrimaryMeasure

Im Vergleich zur Dimension und TimeDimension (vgl. Abbildung 12) wird in PrimaryMeasure die eigentliche Observation (OBS_VALUE) gespeichert. PrimaryMeasures beinhalten üblicherweise Zahlenwerte, weshalb anstelle einer Codelist in dem Beispiel die LocalRepresentation mit dem Typen „Double“ verwendet wird. Wie eine Dimension referenziert auch die PrimaryMeasure auf das entsprechende Concept.

Concept / ConceptScheme

Ein Concept bietet eine voraussetzungsarme Möglichkeit, um beliebige Entitäten (und deren Hierarchien) zu definieren, auf die komplexe Elemente wie Dimensions verweisen können. Es ist ein grundlegendes Element, welches durch eine eindeutige Kombination von Eigenschaften erzeugt wird (vgl. S.39 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)). Im Beispiel sind dies die Übersetzungstexte in Deutsch und Englisch (siehe Abbildung 13). Concepts können wie im vorliegenden Beispiel in ConceptSchemes gruppiert werden. Perspektivisch können ConceptSchemes genutzt werden, um gleiche oder gleichklingende Concepts aus verschieden Bereichen voneinander getrennt zu halten.

Abbildung 13: Beispiel eines SDMX ConceptSchemes bestehend aus drei Concepts

```
<structure:Concepts>
  <structure:ConceptScheme agencyID="UBA" id="CS_COMMON" isFinal="false" version="1.0" >
  <common:Name xml:lang="de">
  common concepts for uba
  </common:Name>

    <structure:Concept id="TIME_PERIOD">
      <common:Name xml:lang="en">year of observation</common:Name>
      <common:Name xml:lang="de">Jahr der Messung</common:Name>
    </structure:Concept>

    <structure:Concept id="CO2_Emission">
      <common:Name xml:lang="en"> source and sink categories </common:Name>
      <common:Name xml:lang="de">Quell und Senkgruppen</common:Name>
    </structure:Concept>

    <structure:Concept id="OBS_VALUE">
      <common:Name xml:lang="en">Observation value</common:Name>
      <common:Name xml:lang="de">erhobener Wert</common:Name>
    </structure:Concept>
  </structure:ConceptScheme>
</structure:Concepts>
```

Quelle: eigene Darstellung, con terra GmbH

Codelist

Eine Codelist ist eine Liste, in der kodierte Werte vorgehalten werden, die dann von anderen statistischen Concepts genutzt werden können (vgl. S. 35 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)). Eine exemplarische Codelist ist in Abbildung 14 dargestellt.

- Im Unterschied zu Concept ist hier der Ausgangspunkt der Entitäten (hier “Codes”) zu beschreiben, die verschiedene Ausprägungen und/oder hierarchische Strukturen enthalten. Eine Reihe von Einschränkungen sorgt dafür, dass diese Strukturen bzw. die Hierarchie, die sie abbilden, simpel bleibt. Die einzelnen Codes werden durch eine eindeutige ID definiert und können Namen und Beschreibungen in verschiedenen Sprachen beinhalten. Insbesondere gilt:
  - Jeder Code kann einen oder keinen Eltern-Code (englisch: parent code) haben
  - Jeder Code kann ein Eltern-Code für beliebig viele Kindelemente sein.
- Komplexere Strukturen können durch Hierarchical Codelists dargestellt werden.

Abbildung 14: Beispiel einer SDMX Codelist

```
<structure:Codelists>
  <structure:Codelist agencyID="UBA" id="CL_CO2_Emission" version="1.0">
    <common:Name xml:lang="en"> Emission source and sink categories -  Codelist</common:Name>
    <common:Name xml:lang="de"> Emissionen: Quell-und Senkengruppen - Codelist</common:Name>

    <structure:Code id="energy">
      <common:Name xml:lang="en"> 1. Energy </common:Name>
      <common:Name xml:lang="de"> 1. Energie </common:Name>
    </structure:Code>

      <structure:Code id="fuel_combustion">
        <common:Name xml:lang="en"> A. Fuel Combustion </common:Name>
        <common:Name xml:lang="de"> A. Verbrennung fossiler Brennstoffe </common:Name>
        <structure:Parent>
          <Ref id="energy" />
        </structure:Parent>
      </structure:Code>

          <structure:Code id="energy_industries">
            <common:Name xml:lang="en"> 1. Energy </common:Name>
            <common:Name xml:lang="de"> 1. Energiewirtschaft</common:Name>
            <structure:Parent>
              <Ref id="fuel_combustion" />
            </structure:Parent>
          </structure:Code>
```

Quelle: eigene Darstellung, con terra GmbH

Hierarchical Codelist

Hierarchical Codelists haben deutlich mehr Freiheitsgrade als Codelists und können deshalb genutzt werden, um noch komplexere Strukturen zu definieren. Eine Hierarchical Codelist beschreibt eine Sammlung von Codes, die in mehreren Eltern-Kind-Beziehungen mit anderen Codes in dem Schema stehen. Hierbei sind eine oder mehrere Hierarchien möglich (vgl. S.95 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)). Hierarchical Codelists werden explizit eingeführt, um in anderen Anwendungen, wie z. B. OLAP Anwendungen, oder anderen Datenvisualisierungssystemen mehrere Sichten auf die Daten bereitzustellen, als es der Fall wäre, wenn lediglich eine simple Codelist verfügbar ist, wie sie für die Definition der DataStructure genutzt wird (vgl. S.92 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)).

Category / Category Scheme

Eine Category ist ein Element auf jeder Ebene einer möglichen Klassifizierung, wie z. B. bei Abschnitten und Unterabschnitten, Gruppen und Untergruppen, Klassen und Unterklassen oder anderen tabellarischen Kategorien (vgl. S.43 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)).

- Ähnlich wie Concepts sind sie in ihrer initialen Struktur simpel und können anderen Entitäten zugeordnet werden, um diese zu beschreiben. Im Gegensatz zu Concepts dient dies weniger einer scharfen Bestimmung anhand von fixierbaren Eigenschaften als einer Einordnung in mehr oder weniger diffuse Themen oder Felder (z. B. “Wirtschaft” oder “Demographie”). Categories sind wie Etiketten zu verstehen, die eine schnell zu erfassende Ordnung herstellen. Auch wenn sie laut Datenmodell einer sehr breiten Menge an Klassen zugeordnet werden können (vgl. S. 42 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)), weist ihr tatsächlicher Gebrauch darauf hin, dass sie eher für

übergeordnete Elemente wie DataFlows und DataSets verwendet werden. Sie helfen, große Datenpools nach Themen zu gruppieren, und erleichtern die Orientierung und Suche in ihnen.

- Analog zum Gebrauch eines ConceptScheme für Concepts werden Categorys mittels eines CategoryScheme gruppiert (vgl. S.42 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)).
- Die Zuordnung einer Entität zu einer Category geschieht mithilfe einer sogenannten Categorisation (S. 43 in (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c)). In Abbildung 15 wird eine Category “Wirtschaft” definiert und mittels einer Categorisation einem DataFlow zugeordnet.

Abbildung 15: Beispiel eines SDMX CategorySchemes

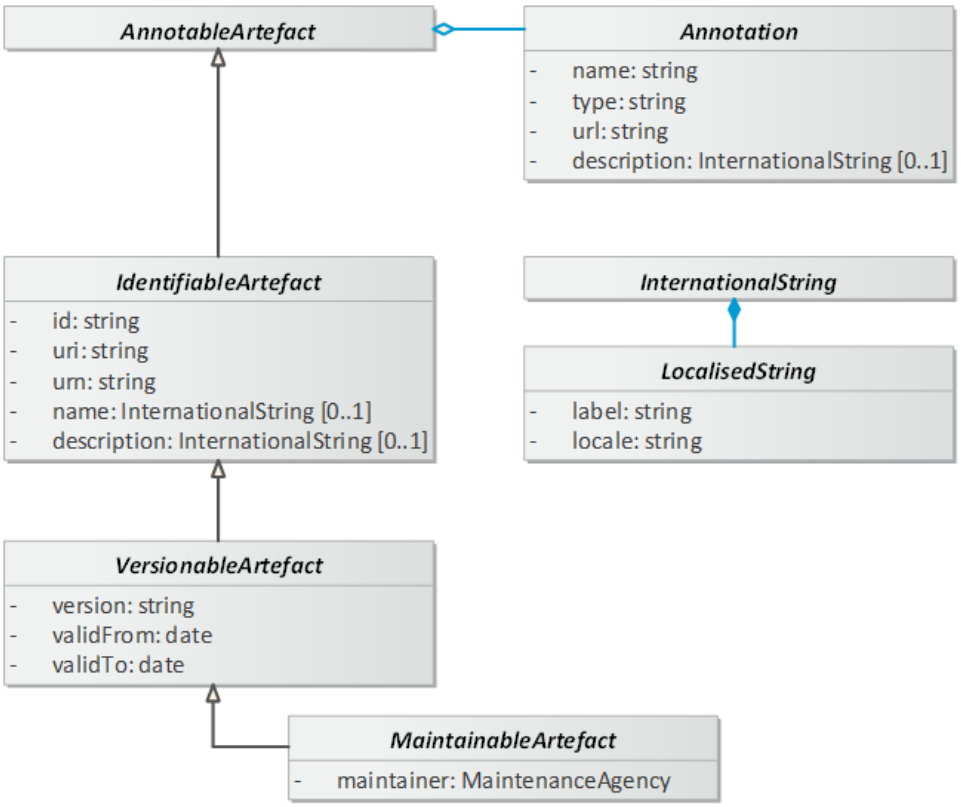
```
<structure:CategorySchemes>
  <structure:CategoryScheme id="CAS_COM_TOPIC" agencyID="UBA" version="1.0"
                            isFinal="false">
    <common:Name xml:lang="en">Topic</common:Name>
    <common:Name xml:lang="fr">Thème</common:Name>
    <common:Name xml:lang="de">Thema</common:Name>
    <structure:Category id="ECO">
      <common:Annotations>
        <common:Annotation>
          <common:AnnotationType>ORenR</common:AnnotationType>
          <common:AnnotationText xml:lang="en">10</common:AnnotationText>
          <common:AnnotationText xml:lang="fr">10</common:AnnotationText>
          <common:AnnotationText xml:lang="de">10</common:AnnotationText>
        </common:Annotation>
      </common:Annotations>
      <common:Name xml:lang="en">Economy</common:Name>
      <common:Name xml:lang="fr">Économie</common:Name>
      <common:Name xml:lang="de">Wirtschaft</common:Name>
    </structure:Category>
  </structure:CategoryScheme>
</structure:CategorySchemes>

<structure:Categorisations>
  <structure:Categorisation id="uba_category" agencyID="UBA" version="1.0"
                            isFinal="false">
    <common:Name xml:lang="en">Categorisation for DF_BOP</common:Name>
    <common:Name xml:lang="de">Kategorieiserung für DF_BOP</common:Name>
    <structure:Source>
      <Ref id="uba_dataflow_id_2" version="1.0" agencyID="UBA"
           package="datastructure" class="Dataflow" />
    </structure:Source>
    <structure:Target>
      <Ref id="ECO" maintainableParentID="CAS_COM_TOPIC"
           maintainableParentVersion="1.0" agencyID="UBA"
           package="categoryscheme" class="Category" />
    </structure:Target>
  </structure:Categorisation>
</structure:Categorisations>
```

Quelle: eigene Darstellung, con terra GmbH

4.1.3 SDMX – Datenstruktur

Im Folgenden werden die wesentlichen Elemente des SDMX-Datenmodells nach (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c) erläutert. In Abbildung 16, Abbildung 17 und Abbildung 18 werden die Objektklassen und ihre Verbindungen auszugsweise in UML-Klassendiagrammen (erstellt mit Enterprise Architect) dargestellt. Der besseren Lesbarkeit halber wird das SDMX-Datenmodell in drei Teilmodelle gesplittet.

Abbildung 16: SDMX–Datenstruktur: Basisklassen



Quelle: eigene Darstellung, hrd.consulting

Das SDMX-Datenmodell kreiert vier Basisklassen, von denen alle wesentlichen Objektklassen abgeleitet werden. Diese vier Basisklassen ziehen sich durch alle Teilmodelle im SDMX Information Model (Statistical Data and Metadata eXchange - SDMX Community, 2011 - c). Die vier Basisklassen sind die Artefacts:

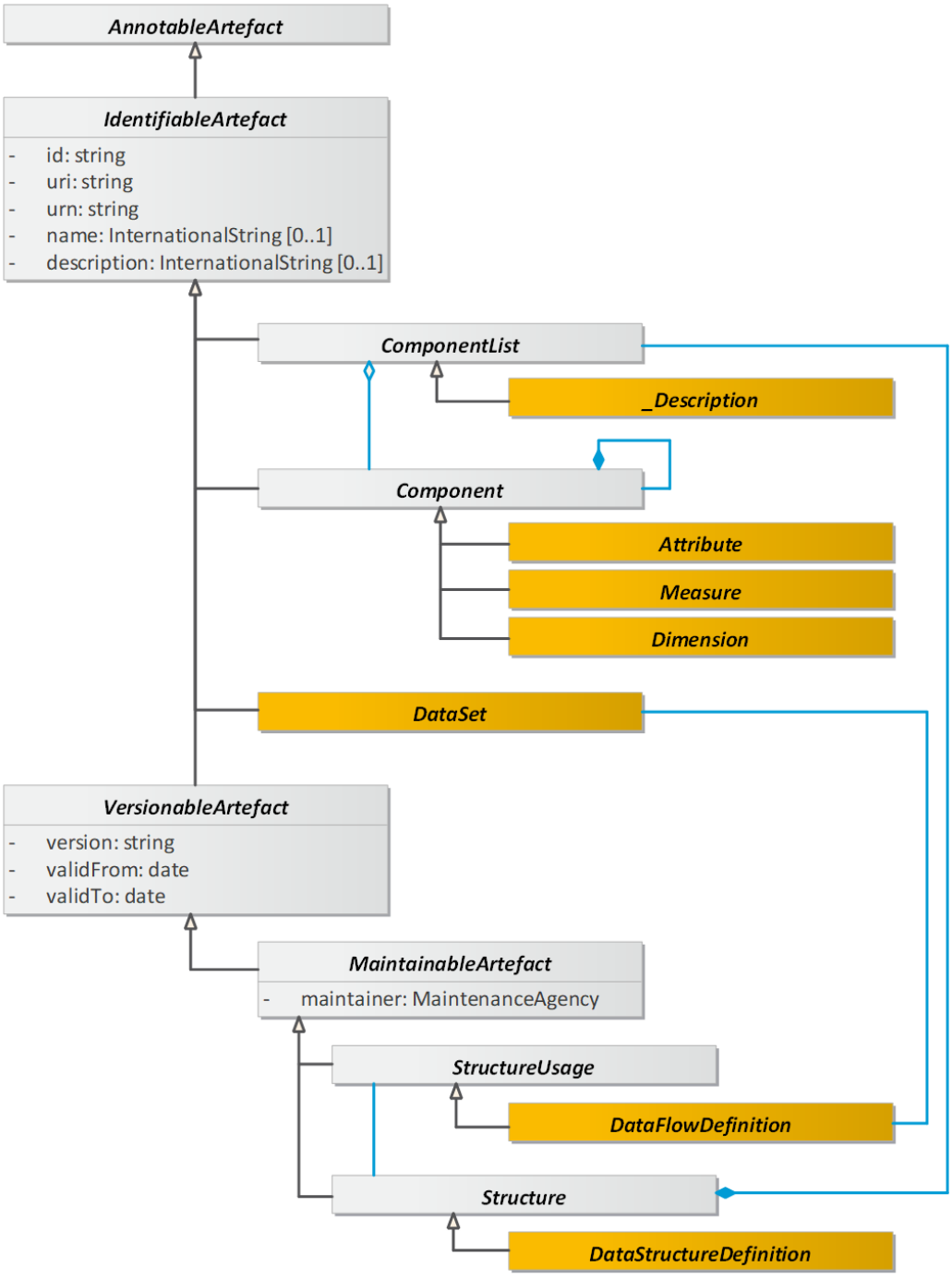
- AnnotableArtefact: mit Kommentar-Attributen (Annotation)
- IdentifiableArtefact: mit Identifikatoren, Links und Textattributen

Die Texte können in mehreren Sprachen (InternationalString) mit Verweis auf die konkreten Texte (LocalisedString) angegeben sein.

- VersionableArtefact: mit Attributen zur Versionierung
- MaintainableArtefact: für die Metadaten mit Angabe der datenpflegenden Stelle (MaintenanceAgency: die konkrete Objektklasse ist von OrganisationRole abgeleitet, vgl. SDMX-Teilmodell in Abbildung 18).

Die Basisklassen setzen entsprechend der Generalisierungsstruktur in der oben genannten Reihenfolge aufeinander auf, sodass die Attribute der jeweiligen Basisklasse von der abgeleiteten Klasse übernommen (geerbt) werden.

Abbildung 17: SDMX–Datenstruktur: DataFlow, DataStructure, DataSet, Component



Quelle: eigene Darstellung, hrd.consulting

In Abbildung 17 sind im unteren Bereich die Objektklassen zur Definition der Strukturen modelliert, die sich von dem MaintainableArtefact ableiten:

- Anwendungsstrukturen (StructureUsage) mit der DataFlowDefinition

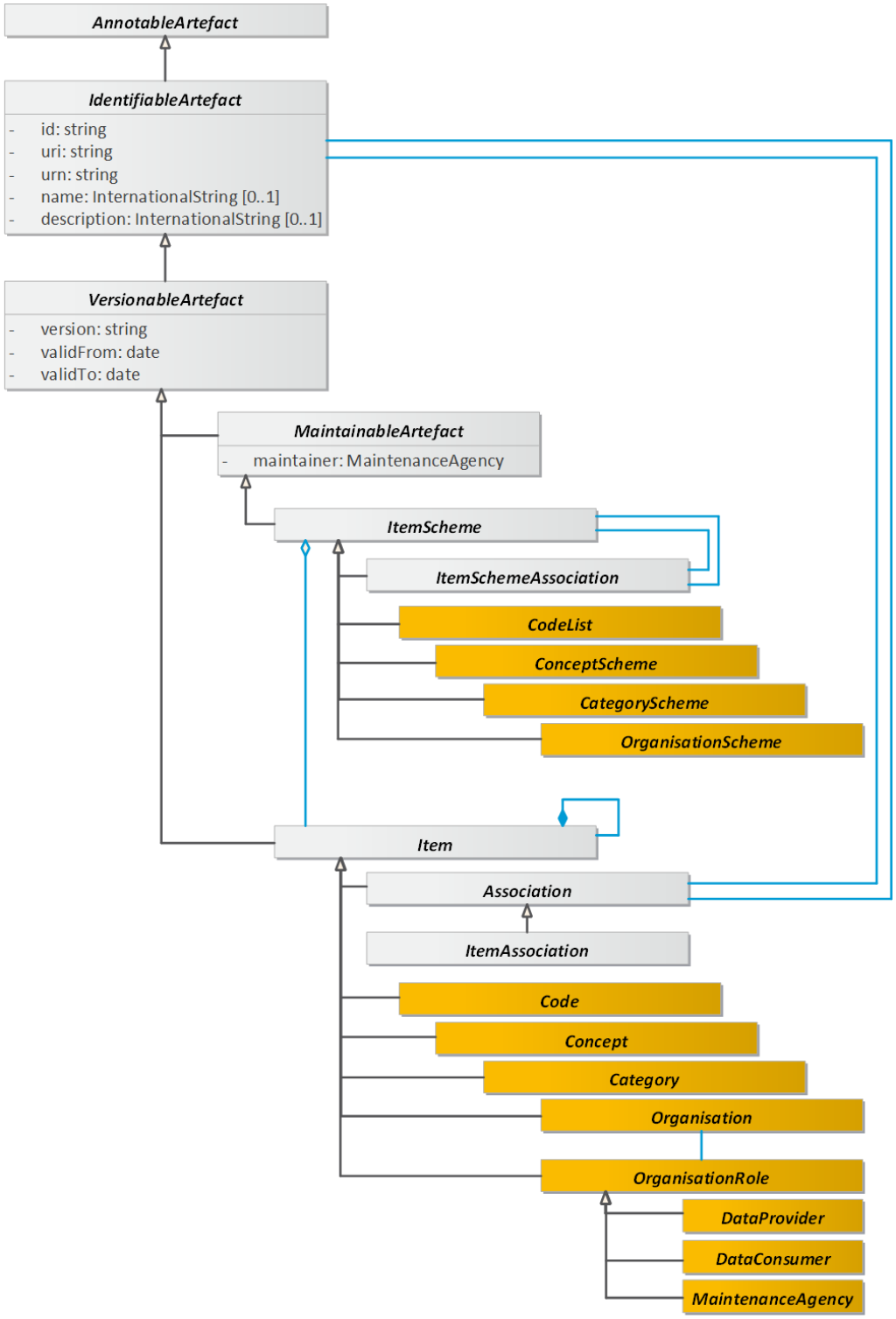
Eine DataFlowDefinition verweist auf ihr DataSet.

- Datenstrukturen (Structure) mit der DataStructureDefinition

Eine DataStructureDefinition verweist auf die ComponentList, was über die Verbindung von Structure zu ComponentList skizziert ist. Die ComponentList enthält die Beschreibungen der Komponenten. Diverse „_Description"-Objektklassen zu den verschiedenen Komponenten sind im Datenmodell von ComponentList abgeleitet. Einer ComponentList sind die einzelnen Komponenten (Component) zugeordnet. Wesentliche Komponenten sind Attribute, Measure und Dimension. Die Komponenten können hierarchisch aufeinander verweisen, was über die Schleifenverbindung bei Component symbolisiert wird.

DataFlowDefinition und DataStructureDefinition können einander zugeordnet werden, was durch die Verbindung zwischen StructureUsage und Structure repräsentiert wird.

Abbildung 18: SDMX–Datenstruktur: Elemente (Item und ItemScheme)



Quelle: eigene Darstellung, hrd.consulting

Das SDMX-Datenmodell folgt stringent der Struktur zur Verwaltung der Elemente (Item) (vgl. Abbildung 18). Die Elemente werden zu Schemata (ItemScheme) zusammengefasst. Da ItemScheme von MaintainableArtifact abgeleitet ist, hat jedes Schema eine datenpflegende Stelle (MaintenanceAgency). Passend zu den Schemen werden die konkreten Elemente über die Verbindung von ItemScheme auf Item zugeordnet. Entsprechende Paare sind:

- CodeList – Code
- ConceptScheme – Concept
- CategoryScheme – Category
- OrganisationScheme – Organisation / OrganisationRole (DataProvider, DataConsumer, MaintenanceAgency).

Über das Element (Item) gibt es eine Schleifenverbindung, sodass die Elemente hierarchisch aufeinander verweisen können.

Im SDMX-Modell ist Association definiert, um beliebige Objekte miteinander verlinken zu können. Das ist durch die zwei Verbindungen (jeweils eine für Quelle und eine für Ziel) von Association auf die Basisklasse IdentifiableArtifact umgesetzt, denn von dieser Basisklasse sind letztendlich alle Objektklassen abgeleitet. Damit erben alle Objektklassen die Verbindung auf Association. Spezielle Associations sind die ItemSchemeAssociation (auf ItemScheme) und die ItemAssociation (auf "Item").

4.1.4 Anmerkung: SDMX 3.0

Im Herbst 2021 wurde SDMX 3.0 als neuer Standard vorgestellt (Statistical Data and Metadata eXchange - SDMX Community, 2021 - a). Eine Übersicht über die Änderungen zur Version 2.1 ist unter (Statistical Data and Metadata eXchange - SDMX Community, 2021 - d) zu finden.

Die .Stat Suite arbeitet aktuell mit SDMX in der Version 2.1. Alle obigen Ausführungen beziehen sich dementsprechend auf SDMX Version 2.1. Wann und ob eine vollständige Umstellung zu SDMX 3.0 geplant ist, ist der Dokumentation der .Stat Suite nicht zu entnehmen.

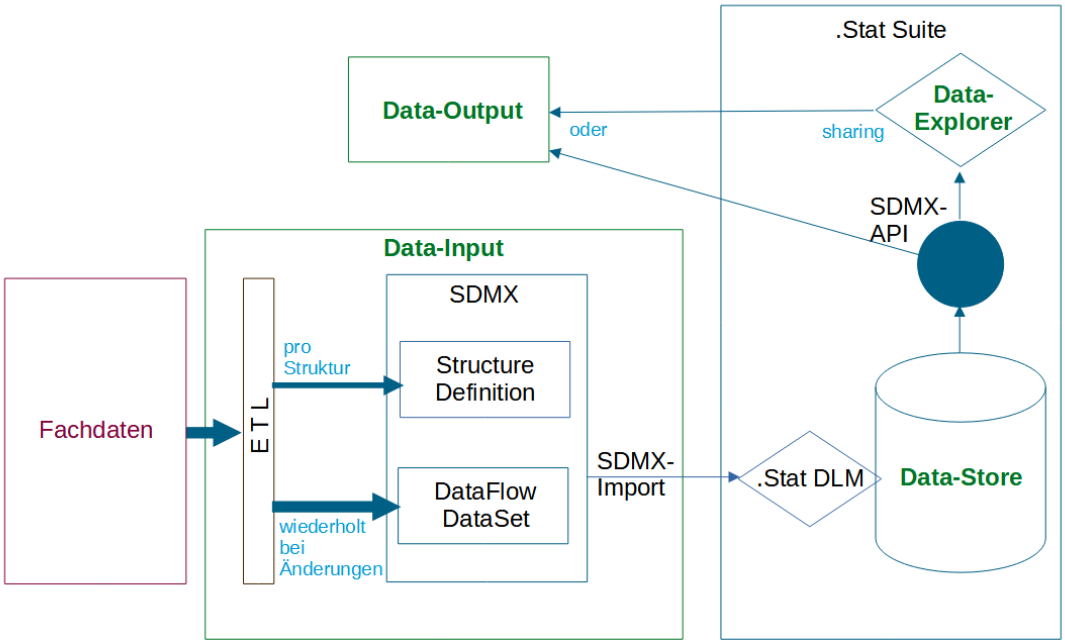
4.2 Redaktioneller Prozess im Kontext von .Stat Suite

Nachdem die grundlegende Funktionsweise der .Stat Suite bereits in Kapitel 3.5.1 beschrieben wurde, soll hier vor allem auf den redaktionellen Prozess eingegangen werden.

In Abbildung 19 sind die Elemente des Lösungsansatzes für den Data Cube skizziert und in Zusammenhang mit den Komponenten der .Stat Suite dargestellt. Die Perspektive folgt den Datenflüssen: Die initialen Fachdaten werden zunächst über ETL-Prozesse in das SDMX-Format konvertiert. Dieses kann über den .Stat Data Lifecycle Manager (DLM) in die Datenhaltung der .Stat Suite eingespielt werden. Der Data-Output kann entweder via der SDMX-API der .Stat Suite oder über die Sharing-Funktion des .Stat Data Explorer erfolgen.

Der Datenfluss sowie daraus resultierende Möglichkeiten für die Redaktion werden im folgenden Kapitel beschrieben.

Abbildung 19: Datenflüsse des Data Cube



Quelle: eigene Darstellung, hrd.consulting

4.2.1 Datenhaltung

Im folgenden Kapitel wird die Behandlung von Daten innerhalb der .Stat Suite genauer beschrieben. Dabei werden zunächst allgemein Konzepte bezüglich Autorisierung und Data Spaces dargestellt. Anschließend werden Importe und Updates von Daten in die .Stat Suite beschrieben.

4.2.1.1 Autorisierung und Data Spaces

Innerhalb der .Stat Suite sind alle SDMX-Strukturen und Daten in sogenannten Data Spaces abgelegt. Sowohl für Daten als auch für Strukturen werden eigene Datenbanken verwendet, um die verschiedenen Elemente getrennt voneinander speichern zu können. Dadurch ist ohne weiteres kein Vermischen der Daten möglich. Spaces können für verschiedene Zwecke verwendet werden. Zum einen können verschiedene Phasen des redaktionellen Prozesses, wie zum Beispiel Datenbereitstellung und Freigabe (siehe Kapitel 2.1), auch auf der Datenebene getrennt werden. Dadurch ist zu jedem Zeitpunkt ersichtlich, welchen Freigabe-Status ein Datensatz gerade besitzt. Darüber hinaus können Spaces verwendet werden, um den Datenzugriff für verschiedene Nutzende zu steuern. Während im Standard alle Daten nur für eingeloggte Nutzende sichtbar sind, können ganze Spaces auch für Gastzugriffe (ohne Login) freigegeben werden. Folgende Spaces erscheinen zum aktuellen Zeitpunkt sinnvoll:

- Datenlieferung
- Qualitätssicherung
- Freigabe
- Freigabe extern

Innerhalb des .Stat Suite DLM können Daten und Strukturen mit wenigen Klicks in einen anderen Space übertragen werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - ag). Dieser Mechanismus kann gut verwendet werden, um die Daten über die verschiedenen Spaces hinweg bis hin zur Freigabe zu publizieren.

Für die konkrete Steuerung der Autorisierung existieren zwei entscheidende Komponenten. Durch die Keycloak Komponente können Nutzende sowie Gruppen verwaltet werden. Nutzende müssen sich einloggen und die Registrierung umfasst den Namen der Nutzenden, Passwort und E-Mail-Adresse. Darüber hinaus können Nutzende in Gruppen zusammengeschlossen werden, um die Autorisierung zu vereinfachen. Diese Einstellungen sind in der Keycloak Administration Console über eine Webansicht möglich.

Für die konkrete Steuerung der Zugangsrechte gibt es derzeit in der .Stat Suite noch keine grafische Komponente. Die Zugriffsrechte müssen stattdessen über die API gesteuert werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - aq). Über den http Post Befehl *AuthorizationRules* können Berechtigungen sowohl für einzelne Nutzende, Gruppen, einzelne Artefakte (zum Beispiel einzelne DataFlows), als auch ganze Data Spaces verwaltet werden. Folgende JSON Konfiguration ermöglicht zum Beispiel den Gast-Zugriff auf den Data Space „Freigabe-Extern“, wobei Permission=3 für die Regeln *CanReadStructuralMetadata* und *CanReadData* verwendet wird:

JSON Autorisierungs-Regel

```
{
    "id": 3,
    "userMask": "*",
    "isGroup": true,
    "dataSpace": "Freigabe-Extern",
    "artefactType": 0,
    "artefactAgencyId": "*",
    "artefactId": "*",
    "artefactVersion": "*",
    "permission": 3
  }
```

4.2.1.2 Einladen neuer Datensätze

Sollen neue Datensätze in die .Stat Suite importiert werden, sind verschiedene Schritte zu durchlaufen. Zunächst muss die Datenquelle analysiert werden. Dabei ist es unter anderem wichtig, die folgenden Informationen zu erfassen (s. Kapitel 3.2.1):

- Datenformat / Datenquelle
  - Für Datenbanken oder APIs sind ggf. Verbindungsparameter und Zugangsdaten abzufragen.

- Zuordnung der Informationen aus dem Datensatz nach: Dimensionen, Attributen, Messwerten
- Identifizierung von Metadaten
- Frequenz der Datenbereitstellung

Im Anschluss müssen die Strukturdefinitionen und die Daten bereitgestellt und gegebenenfalls ins SDMX-Format übertragen werden, falls diese nicht als SDMX-Daten geliefert werden.

4.2.1.2.1 Strukturdefinition

Nach der Analyse des Datensatzes muss in einem weiteren Schritt die Erstellung einer SDMX Strukturdefinition erfolgen (s. Kapitel 4.1). Das Hauptelement für jeden Datensatz ist der DataFlow, welcher unter anderem Namen und Beschreibung des Datensatzes beinhaltet. Diese Informationen werden den Nutzenden später im Data-Explorer angezeigt.

Für die Definition der inhaltlichen Struktur muss dem DataFlow als nächstes eine DataStructure zugewiesen werden. Innerhalb der DataStructure werden alle Dimensionen aufgelistet, die den Datensatz beschreiben. Dabei wird zwischen Dimensions, TimeDimensions und den eigentlichen Messwerten, den MeasureLists, unterschieden. Jede Dimension benötigt eine Referenz zu einem Concept und einer Repräsentation. Das Concept dient zur Beschreibung (unter anderem in verschiedenen Sprachen) der Dimension. Durch die Repräsentation werden mögliche Inhalte durch Codelists oder Facets beschrieben. Codelists beinhalten abgeschlossene Wertelisten, die in einer bestimmten Dimension vorkommen können und erlauben gleichzeitig die mehrsprachige Repräsentation. Durch Facets können in SDMX nicht abgeschlossene Dimensionen durch Datentypen wie Strings, Integer, Double, usw. definiert werden. Das heißt, dass im Voraus nicht alle Werte durch Codelists abgebildet werden müssen. In der .Stat Suite werden zum aktuellen Zeitpunkt jedoch nur Dimensionen mit Codelists unterstützt. Dies gilt jedoch nicht für TimeDimensions und MeasureLists. Zeitstempel können zum Beispiel durch verschiedene vorgegebene FacetTypes wie ObservationalTimePeriod (einfacher Zeitstempel z.B. in der Form YYYY-MM-DD) oder als ReportingYear (Form: YYYY) definiert werden. Messwerte werden auch in Facets zum Beispiel mit dem Typ „Double“ für Gleitkommazahlen angegeben.

Jeder DataFlow verweist auf genau eine DataStructure, aber jede DataStructure kann von beliebig vielen DataFlows verwendet werden. Dadurch kann ein DataFlow wie eine View auf eine Struktur verstanden werden. Zusätzlich zur Beschreibung des Datensatzes durch Kategorien können im DataFlow auch Einschränkungen der Codelists konfiguriert werden. Konkret können einzelne Codes (aber auch Code-Kombinationen) aus der Codelist für den Dataflow explizit erlaubt oder verboten werden. Dadurch können generische Codelists erzeugt werden, die in unterschiedlichen Anwendungsfällen verwendet werden können, wodurch eine Redundanz von Codes über verschiedene Codelists hinweg vermieden werden kann. Die Verwendung solch allgemeiner Codelists wie auch die Erstellung komplexer Datenstrukturdefinitionen, um die Wiederverwendbarkeit in unterschiedlichen Kontexten zu erzielen und potenzielle Redundanzen zu minimieren, wird im Rahmen der folgenden Umsetzungsphase genauer betrachtet. Hierbei sei darauf hingewiesen, dass die Nutzbarkeit von simpleren Strukturdefinitionen und Codelists die Nutzbarkeit erhöhen könnten. Diese Arbeit sollte in enger Zusammenarbeit mit den datenhaltenden Stellen erfolgen und wird in AP6a durchgeführt.

Bei der Definition von Concepts und Codelists sollten die Anforderungen des Datensatzes mit bestehenden Strukturelementen in der .Stat Suite oder mit öffentlichen Elementen aus SDMX Registries (zum Beispiel (Europäische Kommission, o. J. - b)) abgeglichen werden. Die

Verwendung von Registries ist vor allem sinnvoll, wenn internationale Klassifikationen verwendet werden, die auch von anderen Organisationen (zum Beispiel OECD, Eurostat, etc.) in Verwendung sind.

Falls keine vorhandene Codelist verwendet werden soll, muss eine neue definiert werden. Durch Codelists werden alle erlaubten Einträge für einen Datensatz definiert. Hierzu müssen alle möglichen Werte (aktuell vorhandene und zu erwartende) aufgelistet werden. Durch die SDMX Codelists sind auch mehrsprachige Angaben möglich, wobei der in den Daten selbst enthaltende Code für die Darstellung durch die lokalisierten Angaben zum Beispiel in Tabellen oder Diagrammen ausgetauscht wird. Sollte es hierarchische Beziehungen zwischen den einzelnen Elementen geben, so müssen jeweils die Elternobjekte der Codes angegeben werden. Dadurch ist eine beliebig tiefe Verschachtelung möglich. Bei komplexen Hierarchien kann es notwendig sein, eine HierarchicalCodelist zu verwenden. Sollte es keine Liste aller möglichen Werte geben, so kann zum Beispiel ein FME Prozess verwendet werden, um aus dem Datensatz für die jeweilige Dimension alle Unikate herauszufiltern.

Zusatzinformationen (z. B. Fußnoten) die nicht in Codelists abgebildet werden können, können als Attribute (siehe Kapitel 4.1.2.1) hinterlegt werden. Diese werden im .Stat DE als Kommentare direkt an den einzelnen Werten angezeigt (siehe Kapitel 4.2.2.2).

Ist die Strukturdefinition vollständig erfolgt, so kann diese über den .Stat DLM hochgeladen werden. Hierzu wird die Seite „Upload structures“ (Statistical Information System Collaboration Community - SIS-CC, o. J. - au) verwendet. Hierzu muss die XML-Datei hochgeladen und einem Data Space zugeordnet werden.

4.2.1.2.2 Daten

Nachdem die Strukturdefinition abgeschlossen ist, können basierend auf dem DataFlow die Daten vorbereitet werden. Eine einfache Möglichkeit ist das Format SDMX-CSV zu verwenden, da dieses Format, im Vergleich zu den SDMX-XML, durch übliche ETL-Tools, oder sogar durch einfache Excel-Exporte erzeugt werden kann (SDMX Technical Standards Working Group - SDMX TWG, 2021). Tabelle 9 zeigt eine beispielhafte Tabellenstruktur. Durch die Spalte DATAFLOW wird die zuvor definierte Struktur referenziert, wobei „UBA“ in diesem Fall die Agency, „DemoDataflow“ der DataFlow und „(1.0)“ die konkrete Version des DataFlows darstellt. Die Spalte CO2_Emission beinhaltet die Codes aus der jeweiligen Codelist. An dieser Stelle könnten beliebig viele weitere Dimensionen aufgelistet werden. Die Abbildung der Hierarchie-Stufen erfolgt bereits durch die Codelists, weshalb hier eine flache Abbildung der Daten ermöglicht wird. Es folgt TIME_PERIOD als Zeit-Dimension und die konkreten Messwerte in der Spalte OBS_VALUE. Attribute (zum Beispiel Kommentare, Einheiten etc.) werden in SDMX-CSV genau wie Dimensionen behandelt und können als Spalten an Datensätze angehangen werden.

Tabelle 9: Tabellenstruktur in SDMX-CSV am Beispiel von CO2 Emissionen

DATAFLOW	CO2_Emission	TIME_PERIOD	OBS_VALUE
UBA:DemoDataflow (1.0)	energy	1990	989590,091957152
UBA: DemoDataflow (1.0)	industry	1990	59694,7138879148

DATAFLOW	CO2_Emission	TIME_PERIOD	OBS_VALUE
UBA:DemoDataflow (1.0)	energy	1991	881051,945544479
UBA:DemoDataflow (1.0)	industry	1991	55788,3710556744

Die Quelldaten können mit einem der unter Kapitel 4.2.1.4 beschriebenen Tools erzeugt werden. Nachdem die Daten vorliegen, können diese ähnlich wie auch die Strukturen über den .Stat DLM importiert werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - at).

In der Dokumentation der .Stat Suite (Statistical Information System Collaboration Community - SIS-CC, o. J. - c) ist keine API zum automatischen Upload von Daten beschrieben. Da der Upload über den DLM grundsätzlich auch über einen http-Endpunkt erfolgt, ist es denkbar, diesen auch direkt als API ohne die Webseite für automatische Imports zu verwenden. Ob und wie dieser Endpunkt genutzt werden kann, muss während der Implementierungsphase weiter technisch analysiert werden.

Nach dem Upload wird eine E-Mail mit einem Status-Bericht an die Adresse des Nutzenden versendet. Eine kurze Zusammenfassung sowie mögliche Fehler werden in dieser E-Mail bereitgestellt.

4.2.1.3 Updaten und Löschen von Datensätzen

Soll der Datensatz aktualisiert werden, so gibt es mehrere Möglichkeiten. Sobald der SDMX Datensatz erneut hochgeladen wird, werden alle existierenden Werte aktualisiert und neue Einträge werden ergänzt. Daten, die bereits in der .Stat Suite vorliegen aber nicht im Datensatz existieren, bleiben dabei bestehen. Dadurch ist es möglich, einfache Erweiterungen und Aktualisierungen vorzunehmen. Wie beim Upload von neuen Daten, wird auch hier eine Bestätigungsmail versendet.

Soll der alte Datenbestand erhalten bleiben, muss ein neuer DataFlow mit einer neuen Versions-ID erzeugt werden und entsprechend hochgeladen werden. Dieser neue DataFlow kann dieselbe DataStructure referenzieren wie der alte. Durch die Verwendung unterschiedlicher Versionen werden die Datensätze separat behandelt, ohne dass eine komplett neue Strukturdefinition notwendig ist. Im DLM wird im Standard immer nur die letzte Version angezeigt, wobei ältere Versionen über eine Filtermöglichkeit („Filter by Version“) dazugeschaltet werden können. Durch diesen Ansatz können also versionierte Daten und auch zugehörige Datenstrukturen vorgehalten werden.

Um Daten zu löschen, kann entweder der gesamte DataFlow über den .Stat DLM gelöscht oder eine leere SDMX-XML Datei des DataFlow hochgeladen werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - ai). In beiden Fällen werden alle Daten des DataFlows entfernt. Das Löschen von einzelnen Datensätzen, das heißt von einzelne Zeilen oder Zellen aus einer Tabelle, ist aktuell nicht möglich, weshalb für solche Fälle zunächst alle Daten entfernt and anschließend die neuen Daten importiert werden müssen.

Sollte es bei Update-Prozessen zu Anpassungen an den Codelists kommen, so müssen diese ebenfalls aktualisiert werden. Solange nur neue Codes hinzugefügt werden, kann die alte Codelist einfach erweitert werden. Falls Codes angepasst werden und zudem die Codelist auch durch andere Datensätze verwendet wird, so muss eine neue Version der Codelist verwendet

werden. Abhängigkeiten von SDMX-Strukturen können über den .Stat DLM über die Funktion „List related strucutres“ angezeigt werden. Eine neue Version der Codelist macht automatisch eine neue Version des DataFlow notwendig, da dieser die Codelist mit entsprechender Version referenziert und ohne Anpassung daher noch die alte Version der Codelist verwenden würde. Nach Anpassung der Strukturdateien müssen diese erneut über den .Stat DLM hochgeladen werden.

4.2.1.4 Umsetzung

In diesem Abschnitt wird die Überführung der Daten in das SDMX Format beschrieben. Hierzu müssen diese zunächst wie in Kapitel 4.2.1.2 beschrieben analysiert werden. Anschließend erfolgt die Definition der SDMX-Struktur sowie die Konvertierung der eigentlichen Fachdaten. In diesem Abschnitt sollen die Tools SDMX Converter und FME kurz erläutert werden, durch welche die Datentransformationen durchgeführt werden können.

4.2.1.4.1 SDMX Converter

Der SDMX Converter (Europäische Kommission, o. J. - f) ist ein Tool zur Konvertierung und Validierung verschiedener SDMX Formate sowie CSV und Excel. Der Converter kann online frei verwendet werden (Europäische Kommission, o. J. - e), auch wenn dafür ein „European Commissions’s Authentication Service (EU LOGIN)“ (Europäische Kommission, o. J. - d) vorhanden sein muss.

Der Converter kann für Konvertierungen von einfachen Datenstrukturen verwendet werden. Die Voraussetzung ist, dass die notwendigen SDMX Strukturdefinitionen bereits vorliegen. Ein Szenario wäre zum Beispiel, dass neue Datensätze von datenhaltenden Stellen als generische CSV-Datei (noch kein SDMX-CSV) geliefert werden. Diese CSV-Datei könnte mit Hilfe des entsprechenden SDMX-DataFlows in SDMX-XML oder SDMX-CSV überführt werden, welches dann via Upload in die .Stat Suite geladen werden kann.

Die Voraussetzung hierfür ist, dass die CSV-Datei bereits in einer tabellarischen Struktur ist, in der einzelne Dimensionen und Werte durch ihre Spalten zugeordnet werden können. Tabelle 10 zeigt eine solche Tabellenstruktur. Innerhalb des Converters können die Spaltennamen (hier Kategorie, Jahr, Wert) per drag & drop den Zielstrukturen (z. B. Category, TIME_PERIOD, OBS_VALUE) zugeordnet werden. Zusätzlich können einzelne Einträge aus den Quelldaten, falls notwendig, für die durch Codelists geforderten Werte neu zugeordnet werden (z. B. „Kategorie A“ zu „Kategorie_A“). Tabelle 11 beschreibt die gleichen Daten, allerdings in einer Struktur, in der die Dimensionen nicht über die Spaltennamen zugeordnet werden können. Der SDMX Converter ist hier nicht ohne weiteres nutzbar. Einmalig durchgeführte Konfigurationen (z. B. Attributmapping) können exportiert werden und für zukünftige Konvertierungen wiederverwendet werden.

Tabelle 10: Beschreibung einer beispielhaften Tabellenstruktur für Nutzungen im SDMX Converter

Kategorie	Jahr	Wert
Kategorie A	2020	23,4
Kategorie B	2020	5,3
Kategorie A	2021	23,2
Kategorie B	2021	4,2

Tabelle 11: Beschreibung einer beispielhaften Struktur die nicht durch den SDMX Converter konvertiert werden kann

Kategorie	2020	2021
Kategorie A	23,4	23,2
Kategorie B	5,3	4,2

Der SDMX Converter kann also für einfache Datentransformationen verwendet werden, um schnell Daten in ein SDMX Format zu konvertieren, solange die beschriebenen Voraussetzungen erfüllt sind. Eine Automatisierung ist ohne weitere Arbeit nicht möglich. Die zuvor analysierten Excel-Tabellen der verschiedenen datenhaltenden Stellen erfüllen diese Bedingungen nicht, da die Excel-Vorlage über die Spalten fortlaufende Jahresinformationen vorsehen. In Abstimmung mit den datenhaltenden Stellen gilt es in der Umsetzungsphase zu analysieren, in welcher Form die Daten nativ vorliegen.

Weiterhin ist es möglich, SDMX Dateien zu validieren. Dies kann genutzt werden, um potenzielle strukturelle Fehler in den eigenen SDMX Daten zu finden. Die Validierung bezieht sich dabei rein auf die formale Einhaltung der SDMX Strukturen. Es werden keine inhaltlichen Validierungen durchgeführt.

Die konkrete Bedienung des SMDX Converters kann der Bedienungsanleitung (Europäische Kommission, o. J. - g) entnommen werden.

4.2.1.4.2 FME

Während der SDMX Converter (s. Kapitel 4.2.1.4.1) für einfache Konvertierungen verwendet werden kann, sind nicht alle Anforderungen an den Data-Input durch das Tool abgedeckt. Unter anderem sind nur wenige Dateiformate als Quelle unterstützt, welche wiederum bestimmte Vorgaben erfüllen müssen. Es ist jedoch zu erwarten, dass auch Datenbanken / Data Warehouses oder andere Datenformate/ Datenquellen angebunden werden sollen. Zusätzlich sind keine Restrukturierungen innerhalb der Quelldaten möglich. Um zum Beispiel Tabelle 11 in SDMX zu überführen, muss die Struktur des Datensatzes nach ihren Dimensionen aufgelöst werden.

Wie bereits in Kapitel 3.2.4 dargestellt, kann FME als ETL-Tool verwendet werden, um Datentransformationen durchzuführen. Während der SDMX Converter speziell für SDMX Formate entwickelt wurde, ist FME als generisches ETL Tool zu sehen. FME unterstützt verschiedenste Formate, welche zum Einlesen der Daten verwendet werden können (Safe Software Inc., o. J. - b). Die Transformation nach SDMX muss innerhalb von sogenannten FME Workspaces definiert werden. Unter anderem kann für den Export der existierende CSV-Writer verwendet werden, um Daten nach SDMX-CSV zu konvertieren (Safe Software Inc., o. J. - a). Zusätzlich können während der Transformation Daten validiert und transformiert werden (Safe Software Inc., o. J. - c) (Safe Software Inc., o. J. - d).

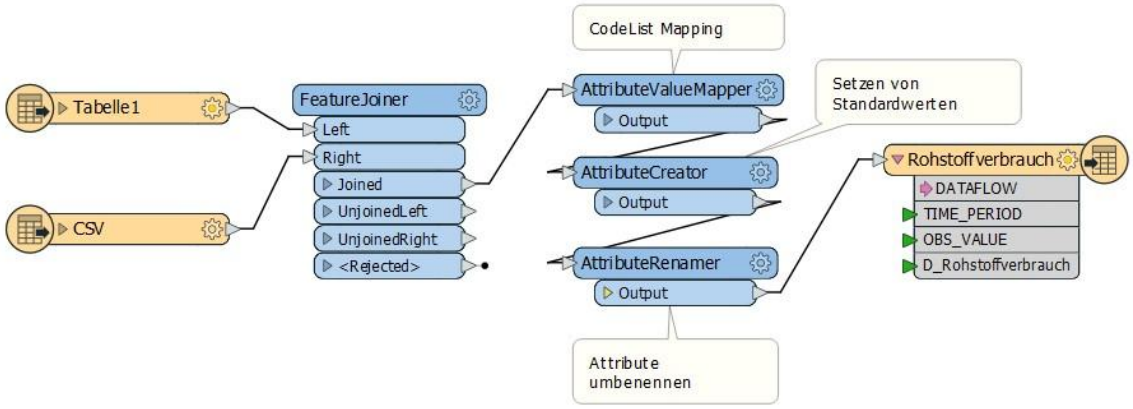
Die folgenden Schritte werden voraussichtlich für die verschiedenen ETL-Prozesse notwendig sein. In Abbildung 20 ist exemplarisch ein simplifizierter FME-Workspace dargestellt, der diese wesentlichen Schritte abdeckt.

- Im ersten Schritt müssen die Daten eingelesen werden. Dies bedeutet, dass zunächst die notwendigen Quellen identifiziert und angesprochen werden müssen. Dies ist für strukturierte Daten (zum Beispiel Datenbanken) verhältnismäßig einfach. Bei Excel-Tabellen kann es notwendig sein, verschiedene Informationen explizit als solche zu identifizieren und entsprechend als Eingabe zu konfigurieren. Beispiele hierfür sind die Auswahl der

notwendigen Tabellenblätter, aber vor allem auch die Definition der Zellen und Zellbereiche, in welchen Daten oder auch Metadaten vorzufinden sind. Diese Schritte sind mit manuellem Konfigurationsaufwand verbunden, können jedoch alle mit den Standard FME-Readern abgebildet werden.

- Anschließend erfolgt bei Bedarf eine Neustrukturierung wie zum Beispiel von Tabelle 11 zu Tabelle 10. In diesem Schritt kann es auch notwendig sein, verschiedene Datenquellen zu verschneiden, um Daten zu denormalisieren und somit die flache Struktur der SDMX-CSV Importdatei zu erzeugen. Diese Strukturierung hängt stark von den Eingangsdaten ab, welche mit jeder datenhaltenden Stelle einzeln abgesprochen werden müssen.
- Da die DataFlow-Dimensionen auf Codelists basieren, kann es weiterhin notwendig sein, Attributwerte zu den definierten Codes umzuschreiben. Dies ist notwendig, da Codes technische Beschreibungen sind, welche erst für die Darstellung der Daten in der jeweiligen Sprache aufgelöst werden. Codes in der .Stat Suite können keine Sonderzeichen, Umlaute oder Leerzeichen beinhalten.
- Zuletzt können weitere Informationen wie der DataFlow-Name ergänzt werden und Attribute zu den Dimensionsbezeichnungen umbenannt werden. Das Schreiben von SDMX-CSV kann über den nativen CSV-Writer von FME durchgeführt werden. Abbildung 20 zeigt einen einfachen Beispielprozess zur Überführung der Daten nach SDMX-CSV. Je nach Komplexität kann der Prozess unterschiedlich ausfallen.

Abbildung 20: Beispiel FME Prozess zu Erzeugung von SDMX-CSV



Quelle: eigene Darstellung, con terra GmbH

Nach der Erzeugung eines FME Prozesses können diese auf dem FME Server bereitgestellt werden. Serverprozesse können für verschiedene Zwecke verwendet werden. Zum einen ist zur Ausführung keine FME Desktop Installation notwendig. Dies bedeutet, dass die Prozesse über einen Webbrowser von einer größeren Personengruppe gestartet werden können. Um die Nutzbarkeit zu vereinfachen, können Prozesse als sogenannte FME Server Apps konfiguriert werden (Safe Software Inc., o. J. - e) (con terra GmbH, o. J.). die Datenstruktur der Quelldaten feststeht und keine ungeplanten Anpassungen zu erwarten sind.

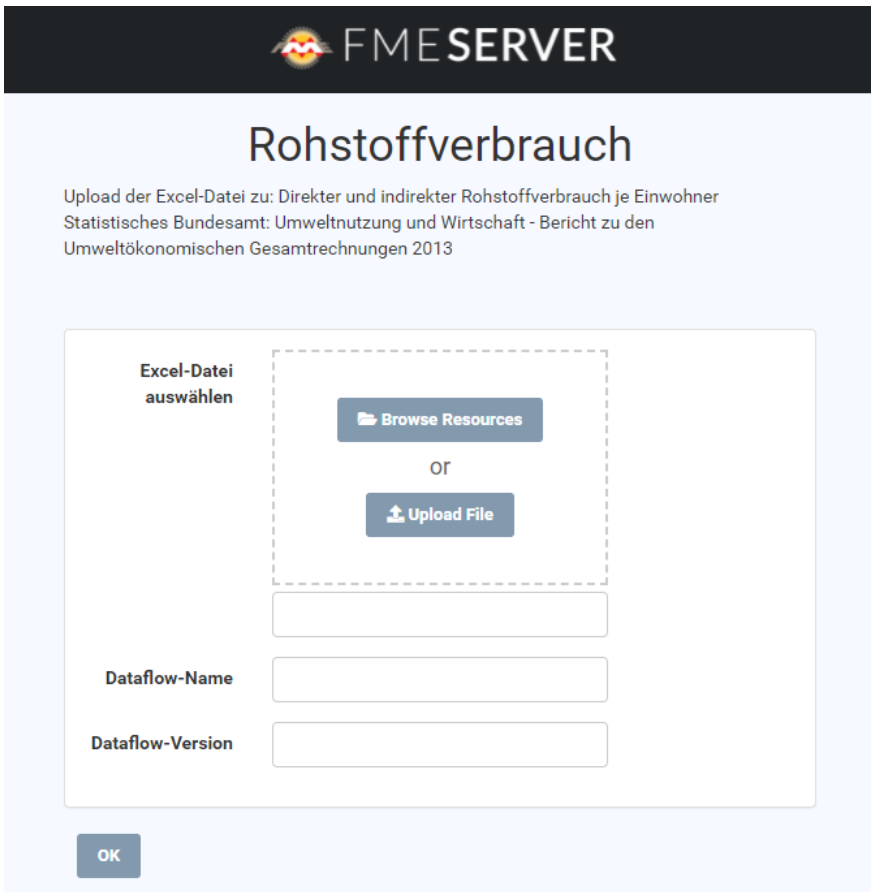
Abbildung 21 zeigt eine beispielhafte Server App, in welcher Nutzende die entsprechenden Informationen (Dateien, DataFlow-Name, Version) bereitstellen können. Parameter wie der DataFlow-Name könnten auch vorkonfiguriert sein. Das Ergebnis des Prozesses kann entweder ein Download der nach SDMX transformierten Dateien oder ein direkter Import in die .Stat Suite sein. Wie in Kapitel 0 beschrieben, muss jedoch noch geklärt werden, wie die Daten automatisch

in die .Stat Suite eingeladen werden können. Die Evaluation wird während der Umsetzung durchgeführt.

Für die Darstellung und Gestaltung der FME Server App können einfache Anpassungen wie andere Farben und ein anderes Banner vorgenommen werden (Safe Software Inc., 2022). Zusätzlich können verschiedene Server Apps in sogenannten Gallery Apps zusammengefasst werden, um eine zentrale Einstiegsseite für ein Fachthema zu ermöglichen (Safe Software Inc., 2020).

Darüber hinaus können Server-Prozesse für zeitgesteuerte Automatisierungen verwendet werden, um z. B. periodische Transformationen zu konfigurieren (Safe Software Inc., o. J. - e). Sowohl Server Apps als auch Automatisierungen sind jedoch nur sinnvoll, falls die Datenstruktur der Quelldaten feststeht und keine ungeplanten Anpassungen zu erwarten sind.

Abbildung 21: FME Server Apps zum Upload von Excel-Dateien



Quelle: eigene Darstellung, con terra GmbH

4.2.2 Datenexploration

Der .Stat Data Explorer (Abkürzung: .Stat DE) stellt die graphische Benutzeroberfläche der .Stat Suite dar, die Datensuche und Datenexploration bereitstellt (Statistical Information System Collaboration Community - SIS-CC, o. J. - i). Im Kontext des Data Cube Projektes ist dies insbesondere für externe Nutzende relevant. In der .Stat DE Applikation stehen die Suche nach Daten, deren Darstellung sowie das Teilen von Daten per Download-Funktion oder über URL-Links im Fokus (Statistical Information System Collaboration Community - SIS-CC, o. J. - n) (Statistical Information System Collaboration Community - SIS-CC, o. J. - a). Die Darstellung der Daten beinhaltet sowohl die Vorschau in Tabellen und Diagrammen als auch die Anzeige von

Kontext- oder Metadaten. Im Weiteren werden die Hauptfunktionalitäten des .Stat DE in den nachfolgenden Unterkapiteln zur Datensuche und zur Datenvisualisierung thematisiert. Hierbei wird die Sichtweise interner und externer Nutzender ebenso wie die Perspektive der (System-)Administration hinsichtlich ihrer Konfigurationsmöglichkeiten berücksichtigt.

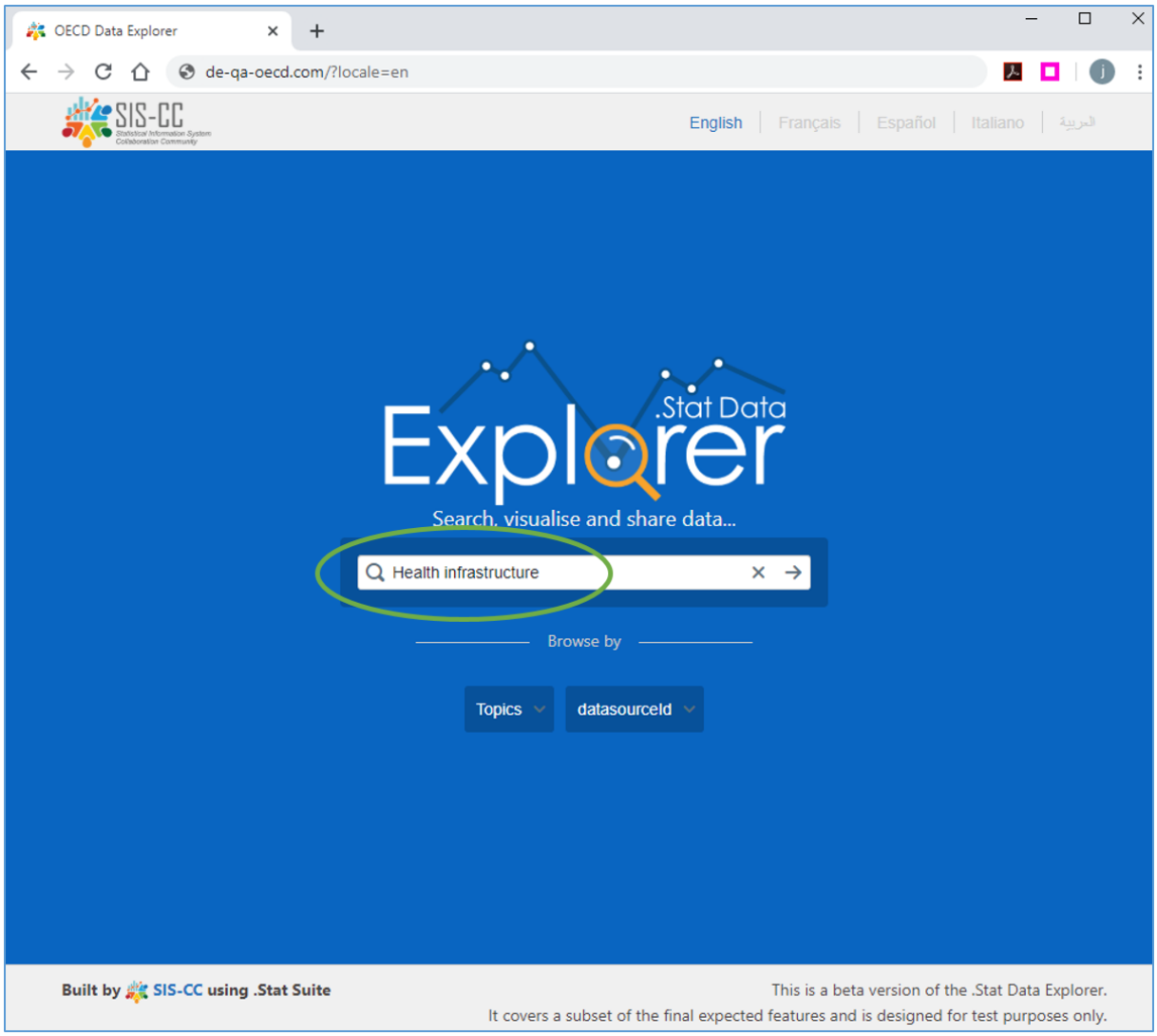
4.2.2.1 Datensuche

Die Startseite des .Stat DE ist der Einstiegspunkt für Nutzende, um Daten zu recherchieren und zu suchen. Um die Suchfunktionen nutzen zu können, ist es wichtig, dass die Daten zunächst indexiert werden. Diese Indexierung sollte durch die Systemadministration durchgeführt werden. Hierbei können nur DataFlows in den Suchindex übernommen werden, für welche bereits Daten importiert wurden. Für weiterführende und insbesondere tiefergehende technische Details bezüglich des Indexierungsvorgangs wird auf die Dokumentationsseite verwiesen (Statistical Information System Collaboration Community - SIS-CC, o. J. - u).

Auf der Startoberfläche des .Stat DE sind ein Freitext-Suchfeld und Auswahllisten für mögliche Filteroptionen (sogenannte Facetten) dargestellt (Statistical Information System Collaboration Community - SIS-CC, o. J. - o). Hierbei ist zu beachten, dass die Freitextsuche und die Filterung über Facetten unabhängig voneinander realisiert sind. Das heißt, dass entweder die Freitext-Suche oder die Suche über eine Facette auf der Startseite abgesetzt werden kann. In diesem Sinne erscheint es empfehlenswert, die Freitext-Suche zuerst zu verwenden und dann im Nachgang das Suchergebnis durch entsprechende Filter zu verfeinern. Mit der Freitext-Suche werden die IDs und die übersetzten Namen der DataFlows, der Kategorien und der Concepts sowie die Codes und Beschreibungen nach dem eingegebenen Suchbegriff recherchiert. Hierbei werden Übereinstimmung zurück geliefert die teilweise oder vollständig zutreffen. Exakte Übereinstimmungen können durch Anführungszeichen um den Suchbegriff erzielt werden. Weiterführende Informationen zu der Freitext-Suche sind in der Dokumentation zu finden (Statistical Information System Collaboration Community - SIS-CC, o. J. - p).

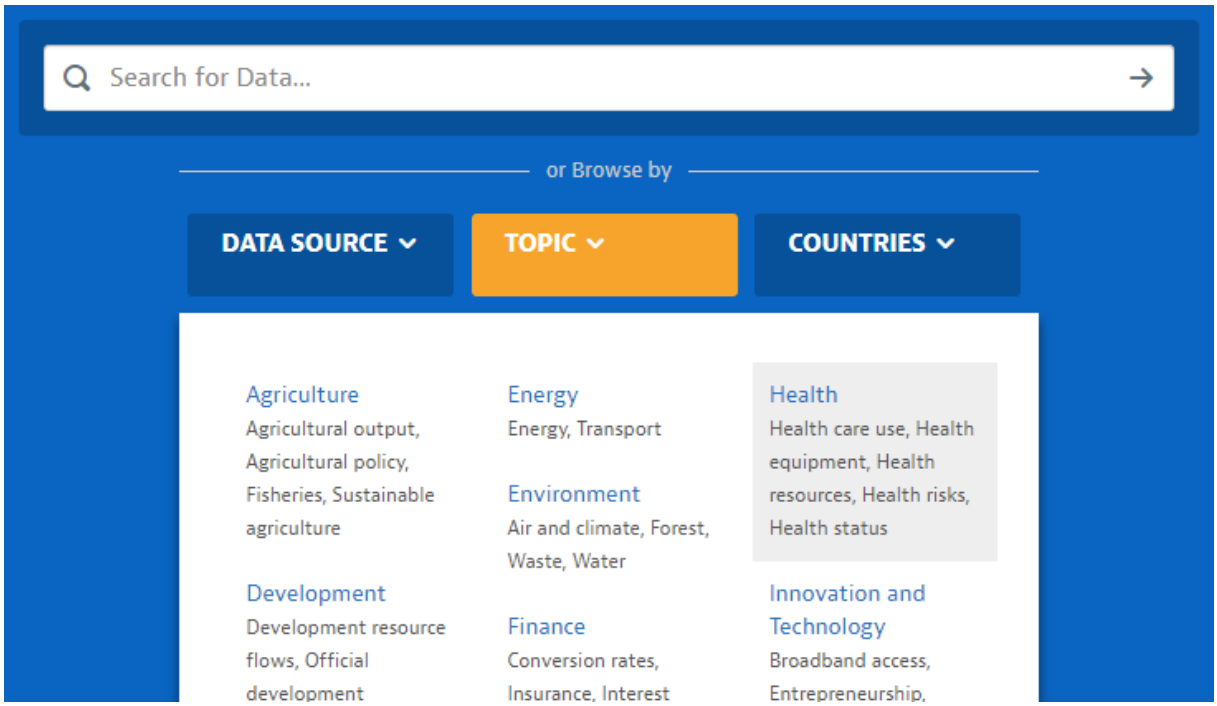
In der Abbildung 22 ist das Freitextfeld mit einem grünen Kreis markiert und beinhalte als Suchbegriff „Health infrastructure“. Die Facetten „Topics“ und „datasourceId“ sind darunter als Dropdown-Listen zur Filterung verfügbar. In den Einstellungen des .Stat DE ist es möglich eine Facette auszuwählen, die standardmäßig mit geöffneter Auswahlliste beim initialen oder erneutem Laden der .Stat DE Startseite dargestellt wird. Weiterhin wird nicht nur die Auflistung der Facettenwerte auf oberster Ebene, sondern ebenso die auf der nächsten Ebene unterstützt (siehe Abbildung 23). Dabei ist zu beachten, dass die Werte der Kind-Ebene erst nach entsprechender Konfiguration in den .Stat DE Einstellungen nutzbar werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - f).

Abbildung 22: Die Startseite des .Stat DE



Quelle: git repository der .Stat Suite documentation (dotstatsuite-documentation/static/images/de-free-text-search.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

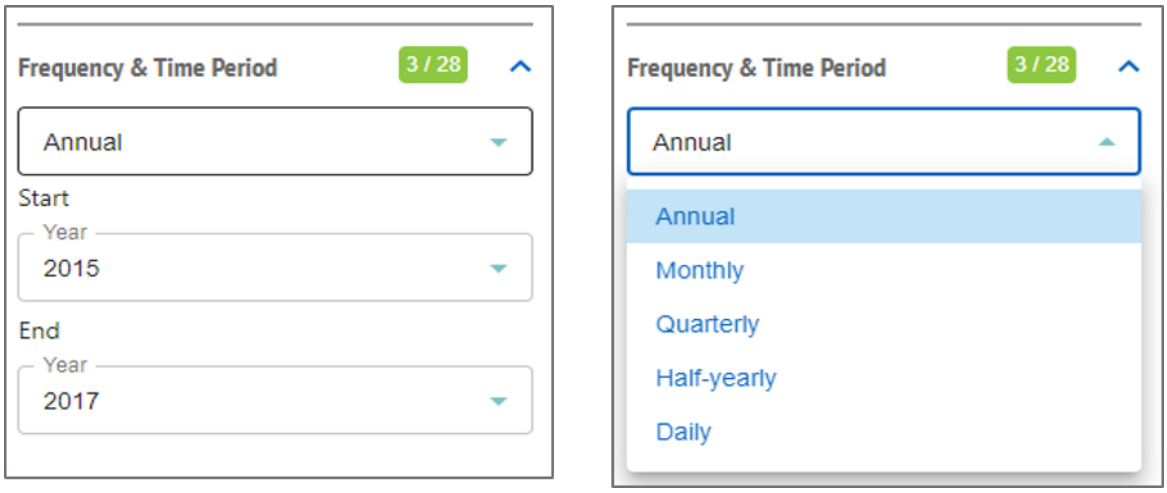
Abbildung 23: Facetten auf oberster und zweiter Ebene auf der Startseite des .Stat DE



Quelle: git repository der .Stat Suite documentation (dotstatsuite-documentation/static/images/de-facet-2.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

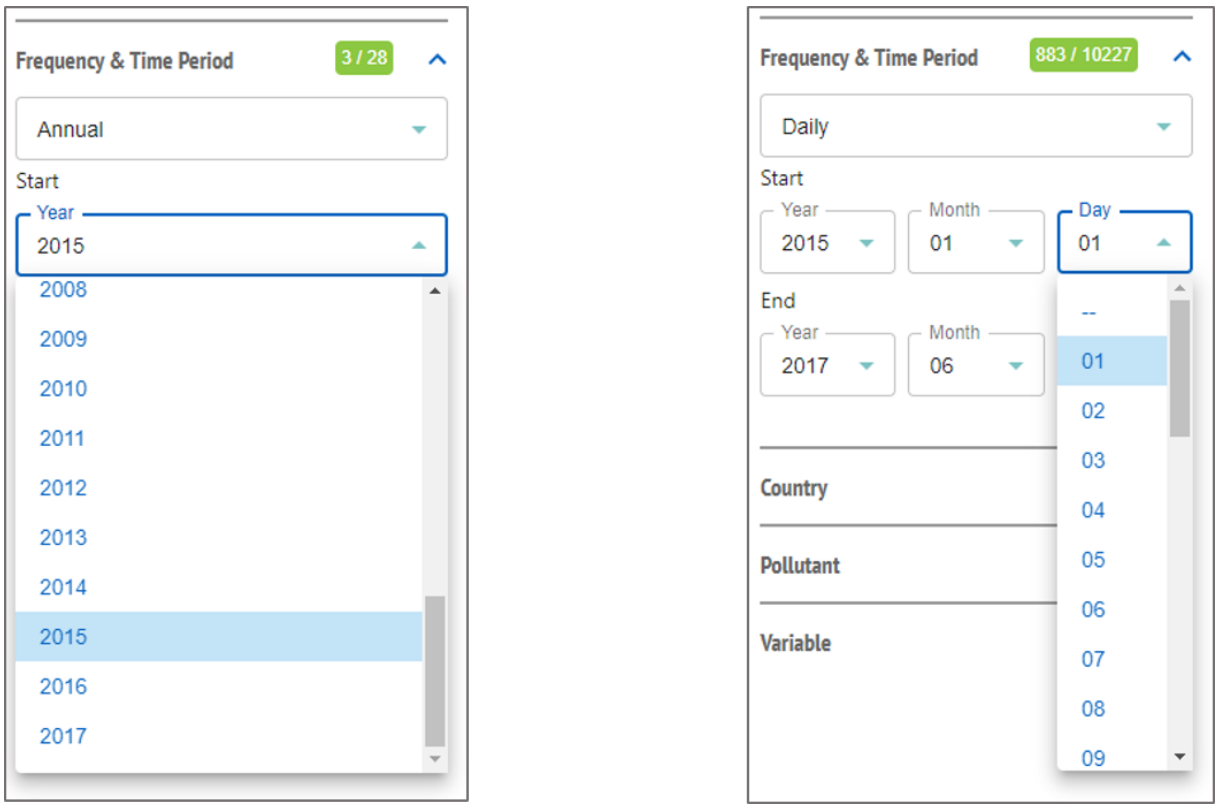
Weiterhin ist es konfigurierbar, ob eine Facette für die Datenquelle erzeugt wird (Statistical Information System Collaboration Community - SIS-CC, o. J. - o). CategorySchemes und die Concepts, die für die DataFlow Dimensionen verwendet werden, werden ebenfalls in der Facettierung berücksichtigt. Hierbei ist zu beachten, dass Concepts, die die gleiche Übersetzung haben, in derselben Facette gruppiert werden. Während im Allgemeinen die gefundenen Werte für die jeweilige Facette zur Filterung bereitgestellt werden, gibt es noch zwei weitere besondere Facetten: Die Zeitperioden-Facette (englisch: Time period) und die Frequenz-Facette (englisch: Frequency). Diese Facetten werden berücksichtigt, falls der DataFlow eine Zeitdimension beinhaltet und werden ansonsten nicht angezeigt. Ebenso können beide oder auch nur die Frequenz-Facette explizit als nicht angezeigt konfiguriert werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - q). Generell erfolgt die Facettierung für die Frequenz, falls zusätzlich noch eine Frequenzdimension vorliegt und diese mehr als einen möglichen Wert enthält. Mögliche Suchwerte für diese Facette sind zum Beispiel jährlich, monatlich oder täglich. Zusätzlich zu der Dokumentationsseite der Facetten in (Statistical Information System Collaboration Community - SIS-CC, o. J. - o) sind für die Facetten für die Zeitdimension in der Dokumentation zur Frequenz und Zeitperiode (Statistical Information System Collaboration Community - SIS-CC, o. J. - q) ergänzende Informationen dargestellt. Während die Frequenz-Facette eine Einzelauswahl einer bestimmten Häufigkeit abbildet, ist für die Zeitperiode ein „Bereichs-Filter“ (englisch: range filter) implementiert, mit dem zwischen einem konkreten Startzeitpunkt und einem Endzeitpunkt gefiltert werden kann (siehe Abbildung 24 und Abbildung 25). Diese konkreten Zeitpunkte können einzeln aus den gesetzten Filtern gelöscht werden. Im Gegensatz dazu kann der Frequenz-Filter nicht gelöscht, sondern lediglich durch eine Alternativauswahl der Häufigkeit geändert werden.

Abbildung 24: Frequenz-Facette mit Dropdown-Liste für Einzelwertauswahl



Quelle: git repository der .Stat Suite documentation (dotstatsuite-documentation/static/images/time-period-1.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

Abbildung 25: Zeitperioden-Facette mit Dropdownlisten zur Auswahl des Start- und Endzeitpunkt in Abhängigkeit zur Frequenz-Selektion



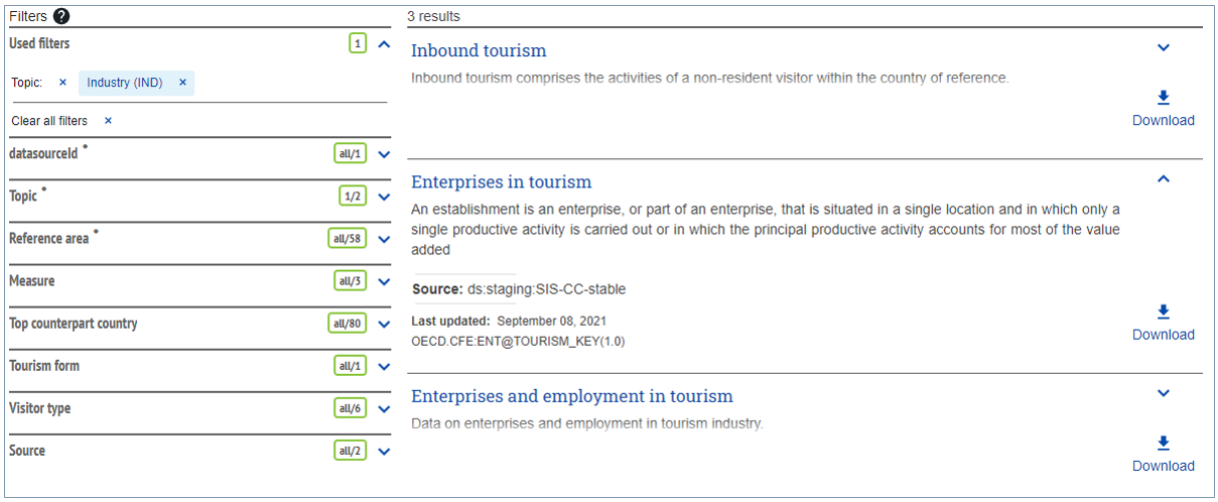
Quelle: git repository der .Stat Suite documentation (dotstatsuite-documentation/static/images/time-period-2.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

Die Abbildung 24 und Abbildung 25 sind bereits Ausschnitte aus der Seite der Suchergebnisse des .Stat DE. Mit Abbildung 26 wird eine Gesamtübersicht dieser Oberfläche ergänzt, auf die Nutzende weitergeleitet werden, nachdem sie eine Suche über die .Stat DE Einstiegsseite (siehe Abbildung 22) abgesetzt haben. In diesem Kontext wird zudem angemerkt, dass gesetzte Filtermöglichkeiten in der URL kodiert werden, so dass eine gefilterte Ansicht per Link geteilt

oder zu einem späteren Zeitpunkt erneut abgerufen werden kann (Statistical Information System Collaboration Community - SIS-CC, o. J. - r). Weiterführende Details zu den URL Parametern sind in Kapitel 4.2.3.1 zu finden.

Grundsätzlich gliedert sich die Seite der Suchergebnisse in zwei Hauptbereiche. In der dargestellten Desktop-Ansicht in Abbildung 26 sind im linken Bereich die Filtermöglichkeiten basierend auf den indexierten Daten und gemäß den konfigurierten und kontextabhängigen Facetten dargestellt. Im rechten Bereich ist die Anzeige der gefilterten Suchergebnisse angeordnet. Die folgenden Beschreibungen basieren maßgeblich auf der zugehörigen Dokumentationsseite für Suchergebnisse (Statistical Information System Collaboration Community - SIS-CC, o. J. - y) und ebenso auf der Dokumentation der Facetten in (Statistical Information System Collaboration Community - SIS-CC, o. J. - o). In der Oberfläche können die Facetten nur einzeln geöffnet werden, doch wird eine Mehrfachauswahl der verschiedenen Facetten unterstützt. Die gesetzten Filter sind in der obersten Facette „Benutzte Filter“ (englisch: „Used filters“) gesammelt und können hier durch die kleinen Schaltflächen mit dem „X“-Symbol wieder deaktiviert werden (siehe Abbildung 26). Die Zahlen in den grünen Kästchen bei den Facetten repräsentieren die Anzahlen, der aktiven und insgesamt verfügbaren Filteroptionen für die jeweilige Facette. Hier ist es wichtig, dass „alle/[Gesamtanzahl]“ (englisch: „all/[total number]“) bedeutet, dass bislang keine Filter aktiv für diese Facette gesetzt wurden und deshalb standardmäßig zunächst „alle“ Möglichkeiten in den Suchergebnissen berücksichtigt werden. Die Reihenfolge der Facetten ist grundsätzlich konfigurierbar. Ebenso ist es möglich Facetten anzuheften (Statistical Information System Collaboration Community - SIS-CC, o. J. - f). Diese werden dann mit einem kleinen Punkt-Symbol markiert (siehe zum Beispiel Facette „Topic“ in Abbildung 26).

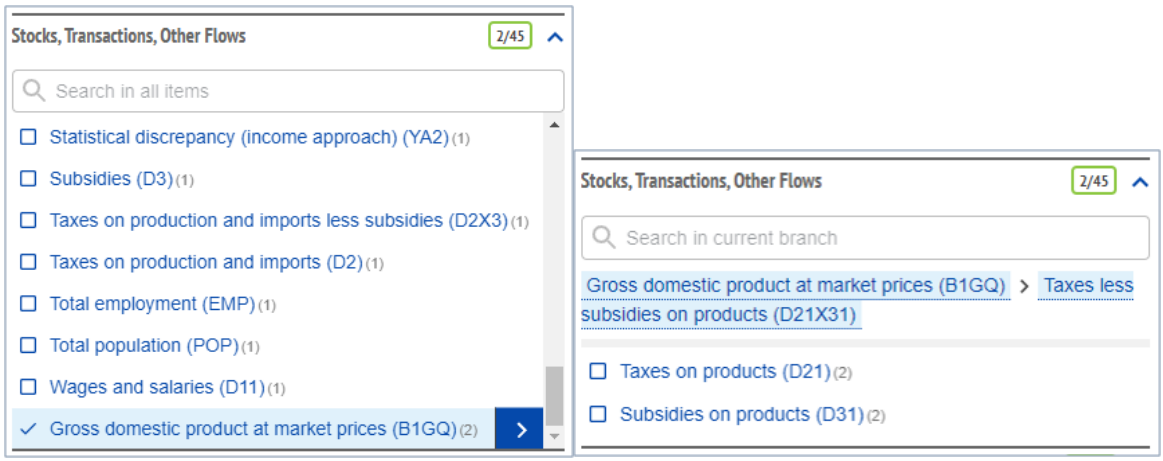
Abbildung 26: Die Seite der Suchergebnisse des .Stat DE



Quelle: git repository der .Stat Suite documentation (dotstatsuite-documentation/static/images/de-result-1.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

In Bezug auf die Facetten wird noch darauf hingewiesen, dass hierarchische Facetten zum Filtern von Suchergebnissen unterstützt werden. In Abbildung 27 ist in der linken Grafik ein Oberbegriff („Gross domestic product at market prices (B1GQ)“) über den blauen Pfeil ausgewählt und in der rechten Ansicht sind die zwei untergeordneten Filtermöglichkeiten („Taxes on products (D21)“ und „Subsidies on products (D31)“) aufgeklappt.

Abbildung 27: Hierarchische Facetten im .Stat DE



Quelle: git repository der .Stat Suite documentation (Grafik links: dotstatsuite-documentation/static/images/de-viewingdata-filters-hierarchicalcontent-rootparents-1.png, Grafik rechts: dotstatsuite-documentation/static/images/de-viewingdata-filters-hierarchicalcontent-rootchildren-1.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

Neben der Facetten-Auswahl im linken Bereich des .Stat DE ist im rechten Bereich die Liste der Suchergebnisse zu finden (siehe Abbildung 26). Hier wird zunächst die Gesamtanzahl der Suchergebnisse dargestellt. Darunter anschließend sind die einzelnen Ergebnisse mit einigen Kontextinformationen dargestellt. In der eingeklappten Darstellung des ersten und dritten Suchergebnisses sind der übersetzte Name des DataFlow als Link und der übersetzte Beschreibungstext zu sehen. In der ausgeklappten Ansicht des zweiten Suchergebnisses werden zusätzliche Informationen aufgelistet, wie zum Beispiel die Quelle (englisch: Source) und das letzte Aktualisierungsdatum (englisch: „Last updated“). Diese Informationen können durch das kleine blaue Dreieck am rechten Rand für jedes Suchergebnis beliebig ein- oder ausgeblendet werden. Ebenso ist am rechten Rand eine Download-Option zum Herunterladen des DataFlow verfügbar, welche explizit hinzukonfiguriert werden kann (Statistical Information System Collaboration Community - SIS-CC, o. J. - f). In dem obigen Ausschnitt der Suchergebnisse in Abbildung 26 sind die Steuerungselemente zum Blättern durch die Liste der Ergebnisse nicht abgebildet. Diese sind unterhalb der Liste am rechten Rand zu finden. Die Navigationselemente beinhalten hier das Vor- und Zurückspringen auf die erste, vorherige, nächste und letzte Seite sowie die Auswahl einer Seite per Freitexteingabe (siehe Abbildung 28). Die Anzahl der dargestellten Ergebnisse pro Seite ist konfigurierbar (Statistical Information System Collaboration Community - SIS-CC, o. J. - f).

Abbildung 28: Navigationselemente für die Vorschauseiten der Suchergebnisse

Quelle: git repository der .Stat Suite documentation (Grafik links: dotstatsuite-documentation/static/images/de-result-2.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

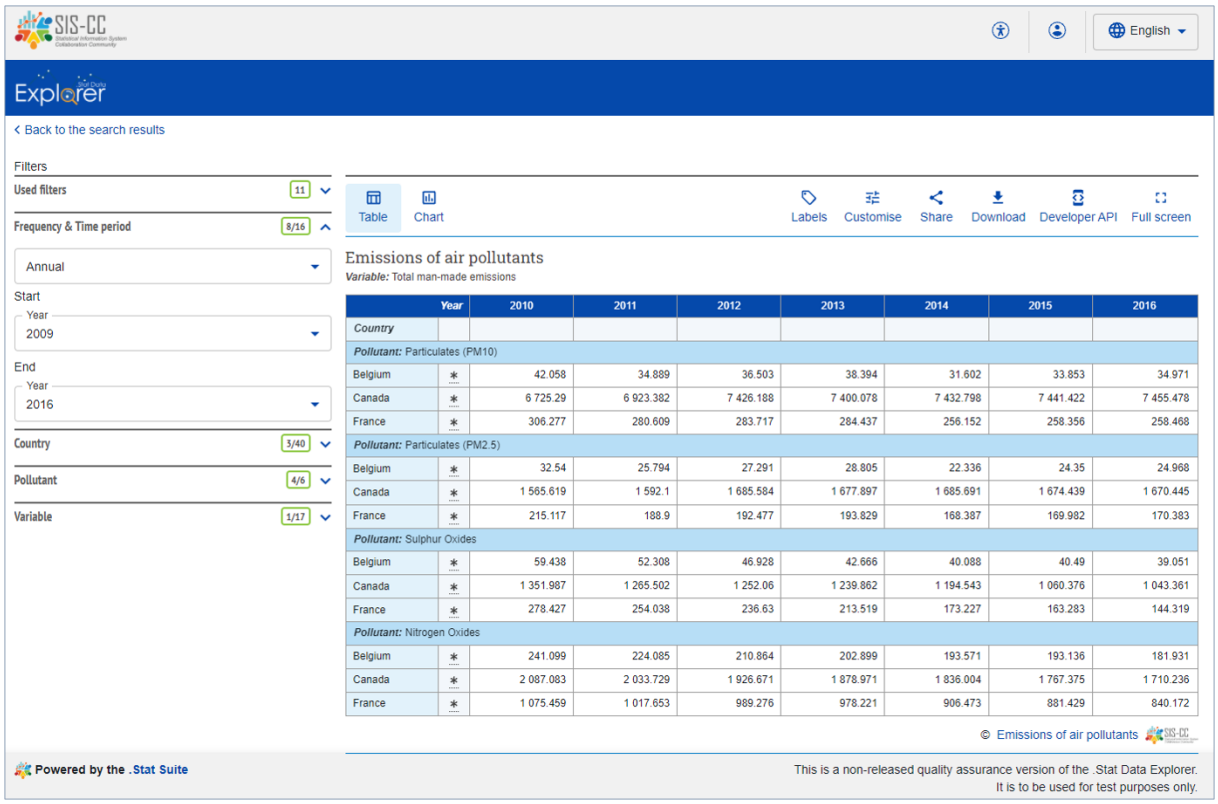
Falls die Freitext-Suche verwendet wird, ist der Suchbegriff in den Suchergebnissen in Gelb farbig hervorgehoben. Dieser und auch Filtereinstellungen in den Facetten werden in der URL kodiert. Dadurch wird ein Teilen oder erneutes Abrufen einer zuvor festgelegten Suche und Filterung ermöglicht. Weitere Informationen zu den verwendeten Kodierungen in der URL sind

in Kapitel 4.2.3.1 und in der Webdokumentation (Statistical Information System Collaboration Community - SIS-CC, o. J. - al) zu finden.

4.2.2.2 Datenvisualisierung

Die folgenden Beschreibungen der Datenvorschau und Datenexploration basieren grundsätzlich auf der zugehörigen Webdokumentation (Statistical Information System Collaboration Community - SIS-CC, o. J. - ac) und den zugehörigen Unterseiten. Der Absprung zu einem konkreten Suchergebnis erfolgt per Verlinkung des DataFlow Namens. Mit diesem werden Nutzende auf die Visualisierungsseite ähnlich der in Abbildung 29 weitergeleitet.

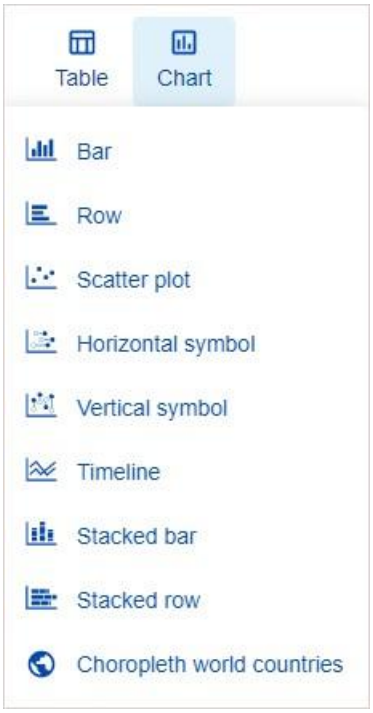
Abbildung 29: Die Datenvorschau im .Stat DE



Quelle: git repository der .Stat Suite documentation (dotstatsuite-documentation/static/images/de-viewing-data.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

In dieser Ansicht sind ebenfalls im linken Bereich Filter-Optionen bereitgestellt, um die abgebildeten Daten zu filtern. Falls man über eine Datensuche zu der Datenvorschau gelangt ist, wird über den Filter-Optionen eine Schaltfläche angezeigt, die zurück zur Seite der Suchergebnisse navigiert (englisch: „Back to the search results“). Im rechten Bereich ist einerseits die Datenvorschau präsent, die standardmäßig als Tabelle visualisiert wird. Zusätzlich ist eine Diagrammansicht verfügbar. Zwischen diesen beiden Ansichten (englisch: „Table“ und „Chart“) kann in der Navigationsleiste im rechten Bereich gewechselt werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - aa). In Abhängigkeit von der Konfiguration des .Stat DE und den vorhandenen Daten sind verschiedene Diagrammvisualisierungen verfügbar. Grundsätzlich können mit der .Stat Suite die in Abbildung 30 gelisteten Diagrammtypen genutzt werden. Für Beispiele zu den einzelnen Diagrammarten wird auf die Dokumentationsseite verwiesen (Statistical Information System Collaboration Community - SIS-CC, o. J. - j).

Abbildung 30: Verfügbare Diagrammtypen im .Stat DE



Quelle: git repository der .Stat Suite documentation (dotstatsuite-documentation/static/images/de-toolbar-chart.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

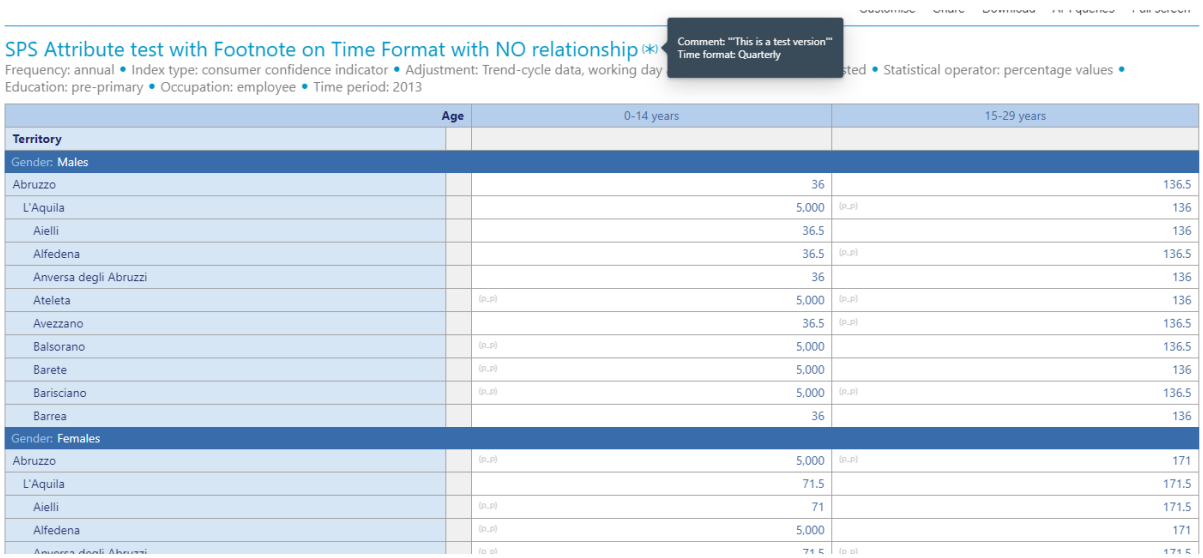
Am rechten Rand wird die Werkzeugleiste, die die Auswahl der Tabellen- oder Diagrammvisualisierung enthält, durch folgende weitere Funktionen als Schaltflächen ergänzt (Statistical Information System Collaboration Community - SIS-CC, o. J. - aa).

- Etiketten (englisch: Labels): Auswahl, ob übersetzte Namen, lediglich IDs oder beides für die vorhandene DataFlow Elemente angezeigt werden.
- Anpassungen (englisch: Customize): Mögliche Anpassungsoptionen in Abhängigkeit des Visualisierungstyps.
  - Für Tabellen können zum Beispiel die Dimensionen den Spalten, Zeilen oder Zeilenabschnitten zugeordnet werden, wodurch eine gewisse Mehrdimensionalität abgebildet wird. Die Anordnung von Zeitperioden in aufsteigender oder absteigender Reihenfolge kann ebenfalls definiert werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - m).
  - Für Diagramme können die Dimensionen in Abhängigkeit des Diagrammtyps zum Beispiel den Achsen oder Symbolen zugeordnet werden. Die Größe des Diagramms in Pixeln ist einstellbar genauso wie die Skalierung der Achsen. Ebenso die Farbauswahl kann verändert werden. Für weitere Konfigurationen wird auf die Dokumentation verwiesen (Statistical Information System Collaboration Community - SIS-CC, o. J. - l).
- Teilen (englisch: Share): Bereitstellung einer URL zum Teilen einer Visualisierungsansicht (Tabelle oder Diagramm), die im .Stat DE akzeptiert und mit einer E-Mail-Adresse bestätigt werden muss. Diese Art der Visualisierungsbereitstellung wird in Kapitel 4.2.3.1 kurz thematisiert und weitere Informationen sind in der Webdokumentation zu finden (Statistical Information System Collaboration Community - SIS-CC, o. J. - z).

- Herunterladen (englisch: Download): Die gefilterten oder ungefilterten Daten der aktuellen Ansicht können im CSV-Format heruntergeladen werden. Ebenso kann die aktuelle Tabellenansicht als Excel-Datei (.xlsx-Format) oder die aktuelle Diagrammansicht als Bild (PNG-Datei) gespeichert werden. Fußnoten und Kennzeichner (siehe Beispiele in Abbildung 29 und Abbildung 31, vgl. Attribute in Kapitel 4.1.2.1) werden in einem weiteren Tabellenblatt gepflegt.
- Entwicklungs-API (englisch: Developer API): Mit der Entwicklungs-API können automatisiert REST-Abfragen für Strukturdefinitionen und Dateninformationen der aktuellen Datenansicht basierend auf dem SDMX 2.1 Format (als XML) generiert werden. Hierbei werden gesetzte Filter- und Visualisierungseinstellungen berücksichtigt. Ergänzende Informationen zu der REST-API der .Stat Suite sind in der Online-Dokumentation zu finden (Statistical Information System Collaboration Community - SIS-CC, o. J. - h). Mittels der REST-Schnittstellen könnten die Daten potenziell auch für andere Fremdanwendungen (z. B. für statistische Berechnungen oder für KI-Auswertungen) abgerufen werden.
- Vollbildschirm (englisch: Full Screen): Mit dieser Option werden die Filteroptionen im linken Bereich ausgeblendet, so dass die Visualisierungen die volle Bildschirmbreite einnehmen können.

Unterhalb dieser Werkzeugleiste befindet sich der Headerbereich der Datenvorschau. Dieser ist für die Tabellen- und die Diagrammansicht identisch und beinhaltet als Titel den übersetzten Namen des DataFlow, falls dieser verfügbar ist und andernfalls die ID. Falls zusätzliche Attribute für diesen DataFlow gepflegt sind, erscheint ein entsprechendes Fußnotensymbol, zum Beispiel ein Sternchen. Diese Fußnoten werden als „Notes“ bezeichnet. Anstelle eines Sternchen-Symbols kann auch eine ID für eine kodierte Fußnote, sogenannte „Flags“, angezeigt werden. Über einen Mouseover-Effekt werden Informationen zu der jeweiligen Fußnote sichtbar (vgl. Abbildung 31).

Abbildung 31: Darstellung einer Fußnote im Headerbereich mittels Mouseover-Effekt



Quelle: git repository der .Stat Suite documentation (dotstatsuite-documentation/static/images/ using-de-footnotes-scenario1-with-no-relationship.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

An dieser Stelle wird nicht detailliert auf mögliche Anwendungsfälle von Fußnoten eingegangen und stattdessen auf die Dokumentation verwiesen (Statistical Information System Collaboration Community - SIS-CC, o. J. - x). Im Untertitel werden die Dimensionen dargestellt, für die entweder nur ein Dimensionswert existiert oder für die durch die aktuelle Filterung nur ein

Wert ausgewählt ist. Sie sind jeweils durch ein Punkt-Symbol voneinander abgegrenzt und falls vorhanden erhalten sie ebenfalls eine zugehörige Fußnote. Im zweiten Untertitel werden die Einheiten (UNIT_MEASURE im SDMX) für die entsprechenden Dimensionen dargestellt, falls welche gepflegt sind. Im Footerbereich unterhalb der Visualisierung kann ein Copyright mit einer Verlinkung zu den Allgemeinen Geschäftsbedingungen sowie ein Logo angegeben werden. Zusätzlich wird eine Verlinkung zu der Standardvisualisierung des aktuellen DataFlow generiert und dargestellt. Die vorgenannten Beschreibungen des Header- und Footerbereichs der Datenvorschau basieren auf der zugehörigen Dokumentationswebseite (Statistical Information System Collaboration Community - SIS-CC, o. J. - k).

Nachdem nun verschiedenste Möglichkeiten für die Datenexploration mit der .Stat Suite beschrieben wurden und in den verschiedenen Kontexten auf Konfigurationsmöglichkeiten eingegangen wurde, werden nun noch weitere optionale Einstellungen für den .Stat DE dargelegt (Statistical Information System Collaboration Community - SIS-CC, o. J. - a) (Statistical Information System Collaboration Community - SIS-CC, o. J. - f). Grundsätzlich ist es möglich beliebig viele Übersetzungen der DataFlow Elemente bereitzustellen und ebenso die Oberfläche entsprechend einer Sprachauswahl darzustellen. Im Kontext der UBA Webseite und speziell der Bereitstellung der Daten zur Umwelt wird es als zielführend erachtet, Lokalisierungen mindestens für die Sprachen Deutsch und Englisch zu pflegen. Ebenso kann die Oberfläche angepasst werden, so dass ein spezielles Corporate Design unterstützt wird. Im Header- und Footer-Bereich der Weboberfläche können dazu Logos dargestellt werden. An dieser Stelle wird darauf hingewiesen, dass im Header-Bereich die Barrierefreiheit aktiviert werden kann (siehe Abbildung 32), um Darstellungen im .Stat DE für Screenreader lesbar abzubilden (Statistical Information System Collaboration Community - SIS-CC, o. J. - r). Im Footer können darüber hinaus noch vordefinierte, übersetzte Texte und Links angezeigt werden (vgl. Abbildung 22 und Abbildung 29). Weiterhin ist die Auswahl der bereitgestellten .Stat DE Module anpassbar, so dass zum Beispiel für externe Nutzende lediglich die Visualisierungsseite erreichbar ist, diese aber nicht eigenständig Datensuchen absetzen könnten. Zudem sei erwähnt, dass die .Stat Suite auf die open-source Bibliothek „Reusable Components for the Web (RCW)“ aufsetzt, die für weitere Implementierungen zur Datenexploration genutzt werden könnte (Statistical Information System Collaboration Community - SIS-CC, o. J. - a).

Abbildung 32: Barrierefreiheit im .Stat DE aktivieren

Quelle: git repository der .Stat Suite documentation (dotstatsuite-documentation/static/images/ de-accessibility1.png in (Statistical Information System Collaboration Community - SIS-CC, o. J. - ad))

4.2.3 Darstellung im CMS für externe Nutzende

Die .Stat Suite bietet verschiedene Möglichkeiten, Daten und Visualisierungen für nicht eingeloggte Nutzende bereitzustellen. In diesem Kapitel werden zuerst Standard-Funktionalitäten beschrieben, anschließend werden Möglichkeiten für native Drupal Integrationen beschrieben.

4.2.3.1 Einbinden über Standard-Funktionalitäten

Das Standard Produkt zur Exploration ist, wie in Kapitel 4.2.2 beschrieben, der .Stat Data-Explorer. Eine Möglichkeit zur Verknüpfung von Drupal und der .Stat Suite sind einfache Verlinkungen. Dazu kann der .Stat Data-Explorer zum Beispiel mit der Startseite verlinkt werden um den Nutzenden die Suche als Einstieg in die Datenexploration zu ermöglichen. Diese Verlinkung könnte ähnlich wie die aktuelle Datensuche direkt unter https://www.umweltbundesamt.de/daten zu finden sein.

Darüber hinaus ist es möglich über die URL des .Stat DE Parameter zu übergeben, um eine Ansicht auf einen bestimmten Datensatz, mit vorausgefüllten Filtern zu erzeugen. Diese URLs können zum Beispiel in Artikeln verlinkt werden, falls Nutzende die Daten-Visualisierung inklusive weiterer Möglichkeiten zur Datenexploration zur Verfügung gestellt bekommen sollen. Die URL ändert sich bei jeder Anpassung der Filteroptionen. Mitarbeitende der Redaktion können also die gewünschte Konfiguration auswählen und anschließend die URL speichern. Eine Beispiel URL sieht folgendermaßen aus:

<Data Explorer URL>/vis?lc=de&df[ds]=freigabe-extern&df[id]=DF_Rohstoffverbrauch&df[ag]=UBA&df[vs]=1.0&av=true&pd=2007%2C2009&pg=0&dq=Biomasse

Die wichtigsten URL Parameter des Beispiels sind in Tabelle 12 beschrieben. Eine detaillierte Beschreibung aller URL Parameter befindet sich in der Dokumentation (Statistical Information System Collaboration Community - SIS-CC, o. J. - al).

Tabelle 12: Beschreibung der .Stat Data-Explorer URL Parameter

Parameter	Bedeutung
lc=de	Sprache = deutsch
df[ds]=freigabe-extern	Data Space = freigabe-extern
df[id]	ID des DataFlows = DF_Rohstoffverbrauch
df[ag]	Agency = UBA
pd=2007%2C2009	Periode = 2007 – 2009
dq=Biomasse	Dimension

Darüber hinaus kann die Sharing-Funktion verwendet werden, um fest Darstellungen zu teilen. In diesem Fall werden die Filter-Parameter und Visualisierungs-Optionen fest gespeichert und können durch den Nutzenden nicht weiter angepasst werden. Zusätzlich kann definiert werden, ob jeweils die neusten Daten, oder ob der Datenbestand als Momentaufnahme verwendet werden soll. Für die Sharing-Funktion muss eine E-Mail-Adresse angegeben werden. Nach der Bereitstellung wird eine Bestätigung mit dem Sharing-Link via Mail bereitgestellt. Zusätzlich wird der HTML-Code zur Einbettung als iFrame angeboten, auch wenn dieser Ansatz für die UBA-Drupal Integration durch Vorgaben des UBA nicht genutzt werden kann. Diese Art von Verlinkung kann in Artikeln verwendet werden, in denen der Text auf die Darstellung und Inhalte Bezug nimmt, wodurch keine dynamische Anpassung gewünscht ist. Sollten Daten/ Visualisierungen nicht als Momentaufnahme außerhalb der .Stat Suite verwendet werden, so müssen diese bei Bedarf nach einer Aktualisierung überprüft werden. Dies ist nicht Teil der .Stat Suite und muss in der Organisation der Redaktion beachtet werden.

4.2.3.2 Native Drupal-Integration

Sollten die zuvor beschriebenen Möglichkeiten nicht ausreichend sein, kann über eine native Drupal-Integration nachgedacht werden. Mit nativer Integration ist an dieser Stelle die Entwicklung eines eigenen Drupal-Moduls gemeint, welches sich nahtlos, ohne iFrames oder Verlinkungen, in das bestehende CMS integriert. Die im Folgenden beschriebenen Möglichkeiten sind denkbare Szenarien.

Der Quellcode für die Ansicht der Sharing-Funktionalität ist open-source (Statistical Information System Collaboration Community - SIS-CC, o. J. - ak) und unter der MIT License zur freien Weiterverwendung als sogenannter .Stat Data-Viewer verfügbar. Um die iFrame Limitierung zu umgehen, ist es denkbar den .Stat Data-Viewer in ein Drupal-Modul zu überführen. Hierzu muss jedoch zunächst überprüft werden, ob die Implementierung grundsätzlich mit Drupal kompatibel ist. Durch die direkte Nachnutzung wären somit ebenfalls fest konfigurierte Diagramme und Tabellen in Drupal möglich. Zusätzliche Interaktionen müssten neu entwickelt beziehungsweise im .Stat Data-Viewer ergänzt werden.

Sollte eine Integration des Data-Viewers in Drupal nicht möglich sein, so können eigene Data-Outputs auf Basis der SDMX-REST-API implementiert werden. Wie in Kapitel 4.2.2.2 beschrieben ist, kann die Entwicklungs-API verwendet werden, um für externe Nutzende freigegebene Daten parametrisiert abzufragen. Der gleiche Mechanismus kommt unter anderem auch bei dem .Stat Data-Viewer und .Stat Data-Explorer zur Anwendung. Grundsätzlich können die Daten beliebig durch andere Anwendungen verwendet werden. Eine komplette Neuentwicklung ist jedoch mit hohem Aufwand verbunden und sollte wenn möglich vermieden werden.

Da die Kompatibilität des .Stat Data-Viewers mit Drupal nicht ohne eine Entwicklungsumgebung sowohl für Drupal als auch für die .Stat Suite sowie mit einer tiefgreifenden Auseinandersetzung mit dem Quellcode möglich ist, soll dieser Schritt im weiteren Verlauf in einer agilen Arbeitsweise durchgeführt werden. Hierzu muss zunächst eine Entwicklungsumgebung etabliert werden, in welcher die Integration getestet werden kann. Sollte die .Stat Data-Viewer Komponente kompatibel sein, so kann die genaue Integration in Drupal beschrieben weiter werden.

4.3 Installation und Betrieb

Die Installation der .Stat Suite sollte nahe an der Installationsdokumentation (Statistical Information System Collaboration Community - SIS-CC, o. J. - c) erfolgen. Grundsätzlich stehen drei Möglichkeiten zur Verfügung:

1. Installation basierend auf dem Quellcode
2. Installation via Kubernetes
3. Installation via Docker

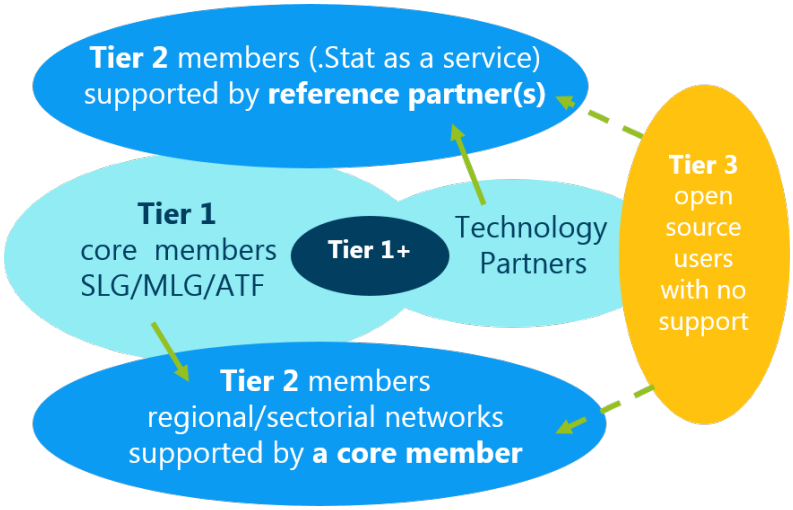
Die genaue Art der Installation kann zu diesem Zeitpunkt nicht definiert werden, da diese nur in enger Abstimmung mit der UBA-IT bzw. den für den Betrieb verantwortlichen Personen erfolgen kann. Anforderungen an die Hardware sind nur für den .Stat Core Data Store definiert (Statistical Information System Collaboration Community - SIS-CC, o. J. - ap) und lauten zum aktuellen Zeitpunkt: 16 GB Arbeitsspeicher, 8 vCPU @ 2.4 GHz und 250 GB Festplattenkapazität.

Im Folgenden wird kurz auf die Installation mittels Docker Compose eingegangen. Hierzu sind im Youtube Kanal der Community SIS-CC (Statistical Information System Collaboration Community - SIS-CC, o. J. - av) die Videos „How to install Stat Suite using docker compose in less than 10 minutes“ (Statistical Information System Collaboration Community - SIS-CC, 2021 - an)

und „.Stat Academy Webinar - .Stat Suite installation using docker compose“ (Statistical Information System Collaboration Community - SIS-CC, 2020) verfügbar. Demnach ist eine Standardinstallation für die Demoversion in weniger als 10 Minuten möglich und der Weg erscheint sinnvoll zum Aufsetzen einer initialen Testumgebung. Laut der der README-Datei des zugehörigen git repository kann diese Installation ebenso als Ausgangspunkt für eine spätere Produktivumgebung genutzt werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - b). In diesem Zusammenhang wird explizit darauf hingewiesen, dass es sich bei dem git repository um eine Demo-Konfiguration handelt. Für eine produktive Nutzung sind weitere Konfigurationen notwendig sind, um für das System eine gewisse Skalierbarkeit zu erzielen und ebenso Sicherungs-/Archivierungsmechanismen zu berücksichtigen. Zudem ist das Thema Sicherheit in der Demo-Version nicht vertiefend behandelt. Zum Beispiel werden für alle Dienste Standard Passwörter verwendet, welche in Konfigurationsdateien im Klartext einsehbar sind.

Es sei an dieser Stelle darauf hingewiesen, dass es externe Firmen, sogenannte Technologiepartner, gibt, die in die Entwicklung der .Stat Suite involviert sind und IT-Dienstleistungen für Nicht-Core Mitglieder der .Stat Suite Community anbieten (Statistical Information System Collaboration Community - SIS-CC, o. J. - s) (Statistical Information System Collaboration Community - SIS-CC, o. J. - as). In diesem Sinne werden technischer Support, Wartungsleistungen und Unterstützung bezüglich Hosting- und Integrationsfragestellungen als kommerzieller Service angeboten. Insbesondere im Hinblick auf die Installation einer Produktivumgebung erscheint es sinnvoll, Unterstützungsleistungen der Technologiepartner anzufragen und langfristig als Zusatzleistung einzukalkulieren, um einen dauerhaft stabilen Betrieb der .Stat Suite im UBA Data Cube zu ermöglichen. Genaue Anfragen bezüglich dieser IT-Dienstleistungen sind nicht Bestandteil des vorliegenden Berichtes, da dies erst nach Abstimmung mit der UBA-IT bezüglich der genauen Betriebsspezifikationen als zielführend angesehen wird.

Abbildung 33: Das Mitgliedschaftsmodell der .Stat Suite



Quelle: Webseite der SIS-CC zum Thema „Governance & funding“ (Statistical Information System Collaboration Community - SIS-CC, o. J. - am)

Im Zusammenhang mit der Möglichkeit, einen Technologiepartner für Zusatzleistungen zu engagieren, wird nun kurz das generelle Mitgliedschaftsmodell der SIS-CC, siehe Abbildung 33, basierend auf den Webseiten zu „Governance & funding“ (Statistical Information System Collaboration Community - SIS-CC, o. J. - am) und zu „Members“ (Statistical Information System

Collaboration Community - SIS-CC, o. J. - ar) der SIS-CC erläutert. Die Kooperation mit einem Technologiepartner wird in die Gruppe „Tier 2“ eingeordnet. Anstelle des Supports durch einen Technologiepartner ist es für Mitglieder dieser Gruppe ebenso möglich, eine Organisation der „Tier 1“ Ebene für Unterstützungsleistungen anzufragen. Die Core-Mitglieder sind in den Entwicklungsprozess der .Stat Suite einbezogen und kofinanzieren die Plattform. Die OECD sei an dieser Stelle exemplarisch als Core-Mitglied genannt. Bevor eine konkrete Anfrage an einen der Technologiepartner gerichtet wird, sollten mögliche Support-Potentiale durch einen Partner wie die OECD eruiert werden. Ebenso sollte in dieser Phase geklärt werden, ob und in welche Höhe Kosten für die Tier 1 und Tier 2 Mitgliedschaften zu erwarten sind. Für Mitglieder der Tier 3 Gruppe werden keine Kosten erwartet, da diese Mitglieder lediglich die open-source Software nutzen ohne jegliche Unterstützungsdienstleistungen. Um die Daten zur Umwelt innerhalb der .Stat Suite sicher verarbeiten und zur Exploration ebenfalls externen Nutzenden bereitzustellen, wird empfohlen eine Mitgliedschaft der Tier 1 oder Tier 2 Gruppe anzustreben. Schlussendlich kann der Antrag auf Mitgliedschaft online gestellt werden und ist über ein Kontaktformular auf der SIS-CC Webseite erreichbar (Statistical Information System Collaboration Community - SIS-CC, o. J. - af).

5 Umsetzung

Im folgenden Kapitel wird die konkrete Umsetzung des in Kapitel 4 dargestellten Lösungsansatzes auf Basis der .Stat Suite beschrieben.

Die Konzeptionsphase endete im Februar 2022 und ging nahtlos in die Umsetzungsphase über. Zeitpunkt des Abschlussberichts ist Ende 2023, wodurch sich ein zeitlicher Sprung von fast zwei Jahren zu den vorherigen Kapiteln ergibt.

Das Kapitel baut sich wie folgt auf: Zunächst wird die Umsetzung der Infrastruktur auf Basis von Docker-Compose beschrieben. Es folgt die Übernahme der Daten verschiedenster Fachbereiche, wobei hier zunächst der Fokus auf die organisatorische Herangehensweise gelegt wird. Danach ist die technische Umsetzung mit FME. Anschließend werden die Entwicklungen zur Integration der .Stat Suite in Drupal beschrieben. Zuletzt wird das Arbeitspaket 6b zur Integration von Metadaten erläutert.

5.1 IT-Infrastruktur

Wie in Kapitel 4.3 beschrieben, sind verschiedene Installationsarten für die .Stat Suite verfügbar. Im Folgenden wird die Bereitstellung der Infrastruktur und die Wahl der Installationsart erläutert. Während der Umsetzungsphase wurde zunächst eine Entwicklungsumgebung innerhalb der Infrastruktur der con terra etabliert. Im Laufe des Projektes wurde diese in die Infrastruktur des UBA migriert.

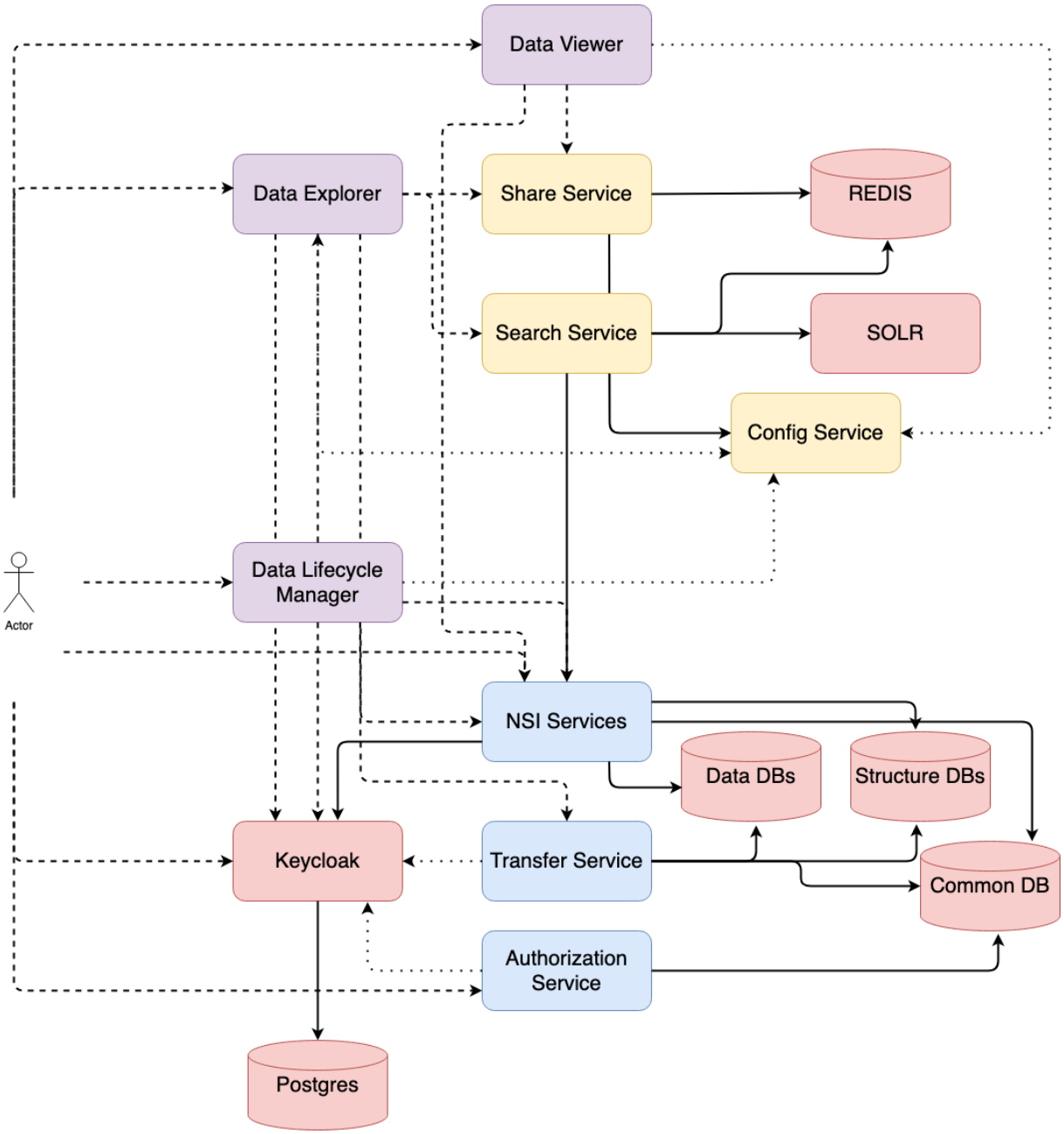
5.1.1 Entwicklungsumgebung bei con terra

Die Infrastruktur der con terra wurde zunächst verwendet, da eine initiale Bereitstellung mit lediglich geringen organisatorischen Hürden möglich war. Zudem ermöglichte diese Art der Bereitstellung Mitarbeitenden der con terra vollen Zugriff auf das System, was vor allem für die initiale Evaluierung verschiedener Installationsvarianten hilfreich war.

Zuerst wurde die Installation der .Stat Suite als native Windows Server Installation durchgeführt. Der Hauptgrund hierfür war, dass zu dem Zeitpunkt wenig Docker oder Kubernetes Erfahrungen im Betrieb des UBA vorhanden waren. Der Betrieb einer "normalen" Windows Umgebung erschien daher organisatorisch einfacher. Die Installation konnte jedoch nicht erfolgreich abgeschlossen werden, da zu dem Zeitpunkt nicht alle .Stat Suite Komponenten in der offiziellen Dokumentation für die Installation vom Quellcode beschrieben waren. Unter anderem war die gesamte Dokumentation des "Authentication Management" noch mit dem Status "to come" versehen. Auf Grund der Vielzahl von verschiedenen Komponenten hatte sich dieser Ansatz jedoch ohnehin als sehr wartungsintensiv dargestellt.

Im zweiten Schritt wurde die Installation via docker-compose durchgeführt. Für die Docker Installation wurde das Betriebssystem zu Ubuntu gewechselt. Die Installation der Demo-Umgebung konnte unter Verwendung des „tachyon“ Releases, wie im git-Repository (Statistical Information System Collaboration Community - SIS-CC, o. J. - v) beschrieben, umgesetzt werden. Die einzelnen Versionen der enthaltenen Systemkomponenten sind ebenfalls im git unter dem besagten Release Tag hinterlegt (Statistical Information System Collaboration Community - SIS-CC, o. J. - aw). Die Installation umfasst eine Vielzahl an Diensten und Komponenten, deren Aufbau in der folgenden Abbildung schematisch dargestellt ist.

Abbildung 34: Komponenten der .Stat Suite



Quelle: https://gitlab.com/sis-cc/.stat-suite/dotstatsuite-docker-compose/-/raw/master/images/OverallArchDiagram.png

Für den öffentlichen Zugriff wurde ein Azure Application Gateway als Reverse Proxy vor der .Stat Suite eingerichtet, um die einzelnen Komponenten anzusprechen. Die folgenden Weiterleitungen wurden eingerichtet:

Tabelle 13: Konfiguration externer Zugriffspunkte der .Stat Suite

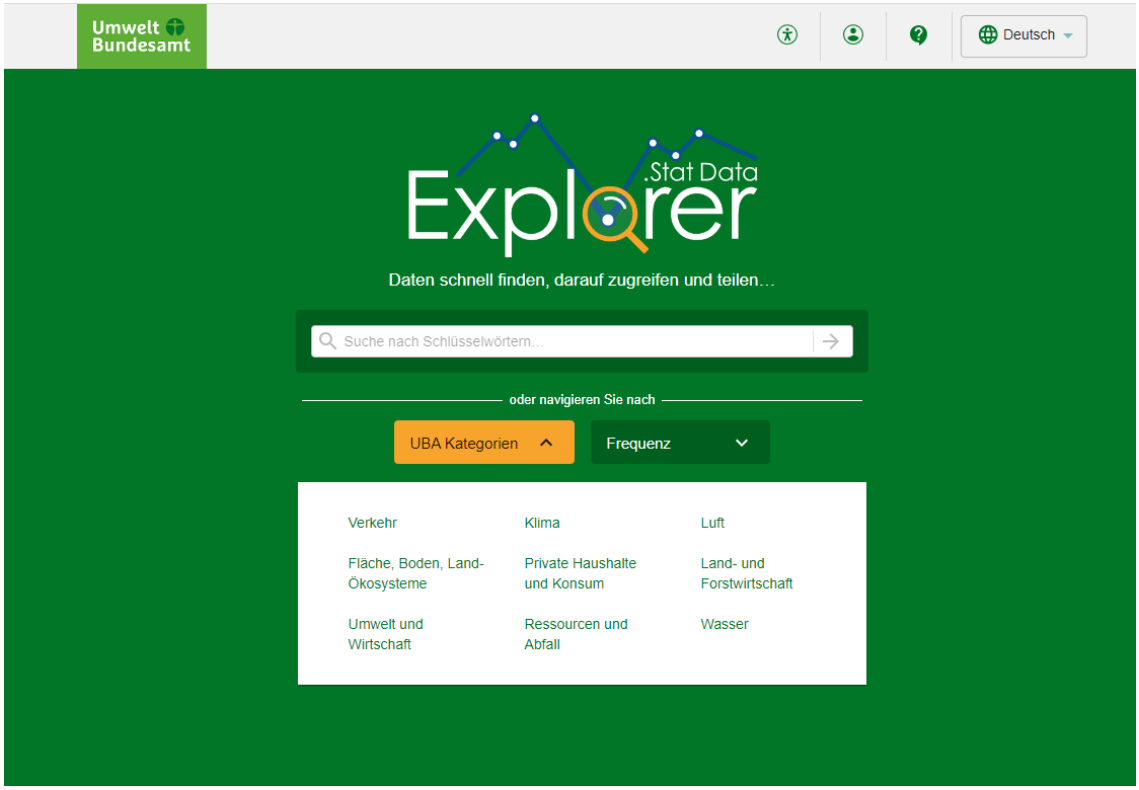
Service	Port	Externe URL
Data Explorer	7001	de-stat.conterra.de

Service	Port	Externe URL
Data Lifecycle Manager	7000	dlm-stat.conterra.de
Data Viewer	7002	viewer-stat.conterra.de
Release Space	81	release-space-stat.conterra.de
Design Space	80	design-space-stat.conterra.de
Transfer Servcie	93	transfer-stat.conterra.de
Auth Service	94	auth-stat.conterra.de
Share Service	3005	share-stat.conterra.de
Search Service	3004	search-stat.conterra.de
Keycloak	8080	keycloak-stat.conterra.de

Die gesamte anonymisierte Konfiguration der docker-compose Umgebung kann im Git Repository des Data Cube eingesehen werden (Statistical Information System Collaboration Community - SIS-CC, o. J. - ah). Eine genaue Auflistung von allen angepassten Konfigurationen findet sich unter *Differences_to_default_demo.md*.

Wesentliche Anpassungen waren vor allem die Anpassung der .Stat Suite an das Cooporate Design des UBA, sowie eine Definition der Kategorien / Facetten auf der Startseite des Data Explorers. Die Anpassungen sind in Abbildung 35Abbildung 35 dargestellt.

Abbildung 35: Darstellung des Data Explorers mit UBA-Farben und Logo



Quelle: eigene Darstellung, con terra GmbH

Zusätzlich zur .Stat Suite wurde in der con terra Infrastruktur testweise die Software Fusion Metadata Registry (Bank for International Settlements - BIS, o.J.) bereitgestellt. Die Software kann zur Verwaltung von SDMX-Elementen in einer webbasierten Oberfläche verwendet werden. Die Vorteile sind daher eine einfache Darstellung und Bearbeitung der Datenstrukturen, ohne dass XML-Kenntnisse erforderlich sind. Die Installation wurde ebenfalls über docker-compose durchgeführt und durch eine Freigabe im Azure Application Gateway im Internet bereitgestellt. Da alle SDMX-Strukturen jedoch automatisiert durch FME-Prozesse erzeugt werden (siehe Kapitel 5.2.2.2) und die Datenstrukturen auch im Git eingesehen werden können, wurde sich gegen die Verwendung der Software entschieden.

5.1.2 Infrastruktur im UBA

Auf Basis der zuvor beschriebenen Entwicklungsumgebung wird derzeit eine Infrastruktur innerhalb des UBA aufgebaut. Auch hier wird ein Linux-Server verwendet, um anschließend die .Stat Suite über Docker-Compose zu betreiben.

5.2 Datenintegration

Im folgenden Abschnitt werden die Arbeiten zur Integration der verschiedenen Daten in die .Stat Suite beschrieben. Dabei wird zunächst die Arbeitsweise und anschließend die technische Umsetzung genauer erläutert.

5.2.1 Beschreibung der Arbeitsweise

Die Beschreibung der Arbeitsweise unterteilt sich zum einen in die allgemeine Organisation durch Meetings und ein Ticketsystem und in die konkrete Umsetzung inklusive der Abstimmung mit verschiedenen datenhaltenden Stellen.

5.2.1.1 Organisation der Arbeitspakete

Für die fachliche Arbeit zwischen con terra und UBA wurde ein wöchentliches Regelmeeting zur Klärung inhaltlicher Fragen etabliert, in welchem zusätzlich zu dem Kernteam auch wechselnde Ansprechpartner der jeweiligen Fachthemen eingeladen wurden.

Darüber hinaus wurde das Gitlab Ticketsystem auf Open CoDe (Komm.ONE, o. J.) gemeinsam mit der Versionsverwaltung verwendet, um einen Überblick über aktuelle Tätigkeiten und die Kommunikation über einzelne Aufgabenpakete zu gewährleisten. Innerhalb des Ticketsystems wurde wie folgt gearbeitet: Einzelne Tickets konnten sowohl von UBA als auch von con terra angelegt und verschiedenen Personen zugewiesen werden. Im Issue Board wurden die Spalten "Open", "In Progress", "Waiting", "Review UBA" und "Closed" angelegt. Neue Tickets werden zunächst in der Spalte "Open" angelegt. Für jeden Dataflow wird initial ein Ticket angelegt und sobald alle Daten vorliegen, Mitarbeitenden der con terra zugewiesen. Sobald die Bearbeitung beginnt, werden diese in die Spalte "In Progress" geschoben. Aufgaben, die gerade auf externe Informationen warten, finden sich in der Spalte "Waiting". Gründe hierfür sind zum Beispiel fachliche Rückfragen, die direkt durch die verschiedenen datenhaltenden Stellen geklärt werden müssen. Durch die Kategorisierung ist somit klar, dass an der jeweiligen Aufgabe gerade nicht weitergearbeitet wird. Tickets, die von con terra bereits abgearbeitet wurden, landen zunächst in "Review UBA" und werden erst nach Überprüfung nach "Close" überführt. Durch die Kommentarfunktion können fachliche Fragen direkt zugehörig zu dem jeweiligen Aufgabenpaket diskutiert und gleichzeitig dokumentiert werden.

Zusätzlich wird Git als Software zur Versionsverwaltung verwendet. Hierbei wird parallel zum Ticketsystem auf die gehostete Variante von OpenCoDe zurückgegriffen Die genaue Ordnerstruktur ist in Kapitel 5.2.2 genauer erläutert.

5.2.1.2 Organisation der Datenintegration

Die operative Integration der Datensätze datenhaltender Stellen lief iterativ während der gesamten Umsetzungsphase des Projektes ab. Ziel war es mehrere Datensätze eines Themas (zum Beispiel Verkehr, Luft, Wasser, …) zu bearbeiten während gleichzeitig Abstimmungen für das nächste Thema durchgeführt wurden, um eine möglichst kontinuierliche Arbeitslast zu erreichen.

Die Priorisierung, Kontaktaufnahme und initiale Abstimmung mit den datenhaltenden Stellen wurden direkt durch Mitarbeitende der Redaktion durchgeführt. Nach der Identifikation möglicher Datensätze für ein bestimmtes Thema erfolgt jeweils, wie in Kapitel 4.2.1.2 beschrieben, die Definition der Dimensionen und der jeweils verwendeten Codelists jedes Dataflows. Dieser Prozessschritt benötigt eine enge Abstimmung von allen Beteiligten, da die Definition der Codelists fundierte inhaltliche Kenntnisse des jeweiligen Datensatzes als auch mit bestehenden Strukturen der .Stat Suite voraussetzen. Nur durch Wiederverwendung bestehender Codelists über die verschiedenen Themen hinweg, ergibt sich ein homogener Datenbestand. Ein Beispiel für eine solche gesamtheitliche Betrachtung sind Luftschadstoffe wie SO_2 oder CO. Diese könnten in einer themenspezifischen Codelist definiert werden die nur für das Thema Luft benötigt wird. Sinnvoller ist jedoch die Verwendung einer allgemeingültigen Codelists "Substances", die über viele Dataflows hinweg anwendbar ist.

Für einen besseren Überblick wird eine Dataflows Matrix verwendet (siehe zum Beispiel Abbildung 36) um alle Dataflows eines Themas aufzulisten und die jeweils zu verwendenden Codelists zu definieren. Die weitere Verwendung der Matrix wird in Kapitel 5.2.2 beschrieben. Die Matrix ermöglicht eine Strukturdefinition, ohne dass zu tiefes SDMX-Wissen vorausgesetzt wird und dient als Mittel, um mit den datenhaltenden Stellen in den Austausch zu kommen.

Abbildung 36: Dataflow Matrix nach Thema

Technischer Name	Version	C_AREA	C_FEDERAL_STATES	C_COMMUNITIES	C_FREQUENCY	C_UNIT	C_SUBSTANCES	(
DF_AIR_QUALITY_AGGLOMERATIONS	1.1	1.0	0	0	1.0	1.0	1.1	(
DF_AIR_QUALITY_TRENDS	1.1	0	0	0	1.0	1.3	1.1	(
DF_AIR_QUALITY_EXCEEDANCE_PM10	1.1	0	2.0	0	1.0	1.3	1.1	1
DF_AIR_QUALITY_EXCEEDANCE_O3	1.1	0	2.0	0	1.0	1.3	1.1	1
DF_AIR_QUALITY_EXCEEDANCE_NO2	1.1	0	2.0	0	1.0	1.3	1.1	1
DF_AIR_EMISSIONS_INDEX	1.1	0	0	0	1.0	0	1.1	(
DF_AIR_EMISSIONS_TRENDS	1.1	0	0	0	1.0	1.3	1.1	(
DF_AIR_DEPOSITION_HEAVY_METALS	1.1	1.0	0	0	1.0	1.0	1.1	1
DF_AIR_DEPOSITION_BIO	1.1	1.0	0	0	1.0	1.0	1.1	(
DF_AIR_DEPOSITION_IONS	1.1	1.0	0	0	1.0	1.0	1.1	1

Quelle: eigene Darstellung, con terra GmbH

Die Dataflow Matrix wird mit verschiedenen anderen Konfigurationsdateien zur Beschreibung der Metadaten im Git Repository eingecheckt. Details zu Konfigurationsdateien und der Dateistruktur im Git Repository finden sich in Kapitel 5.2.2. Sollten die jeweiligen Quelldaten für einen Dataflow dateibasiert vorliegen, werden diese ebenfalls von Mitarbeitenden des UBA im Git Repository hochgeladen. Eine finale Dateiablage separat vom Git Repository wird derzeit noch erarbeitet. Dies ist unter anderem notwendig, da einige Quelldaten nicht mit dem Rest der Prozesse veröffentlich werden können. Darüber hinaus ist Git nicht zur Verwaltung großer Datenmengen optimiert und was langfristig, bei wachsendem Datenbestand, zu Performance-Problemen führen würde.

Zur besseren Organisation werden Arbeiten in Git durch das UBA auf separaten Branches durchgeführt und nach Fertigstellung als Merge request (GitLab Inc., o. J.) an con terra übergeben. Dies ermöglicht ein paralleles Arbeiten an Dataflows, ohne bereits produktive Dataflows, oder anderen Arbeitenden zu beeinflussen. Merge requests werden von con terra bearbeitet und in den Hauptbranch gemerged. Sobald ein Merge request gestellt wird, werden die entsprechenden Mitarbeitenden automatisch benachrichtigt und die Anpassungen können bearbeitet werden. Nach Abschluss der Arbeiten von con terra wird der entsprechende Mitarbeitende des UBA durch Zuweisung des Tickets (Spalte "Review UBA") informiert.

5.2.2 Technische Umsetzung

Die technische Umsetzung zur Erstellung von SDMX-Strukturen und Übernahme der zugehörigen Daten spielt sich vor allem im Git ab, wobei FME zur Umsetzung der einzelnen Prozessschritte verwendet wird. Im Folgenden wird zunächst die Dateistruktur des Git erläutert, anschließend werden die verschiedenen Prozesse zur Umsetzung genauer beschrieben.

5.2.2.1 Git Dateistruktur

Im Git werden FME-Prozesse, SDMX-Strukturen und Dataflows abgelegt. Die wesentliche Dateistruktur sieht wie folgt aus:

- cube
  - concept_overview.csv
  - Je ein Ordner pro Codelists mit

        - output
        - elements.csv
        - metadata.csv
- dataflows
    - Je ein Ordner pro Thema (air, climate, …) mit
        - Overview_matrix.csv
        - Je ein Ordner pro Dataflow mit
            - output
            - data
            - transformation
                - hints.txt
                - main_process.fmw
            - metadata.csv
- transformations

5.2.2.1.1 Concepts und Codelists

Im Ordner Cube werden die SDMX-Elemente Concept und Codelist definiert. Die SDMX-Strukturen werden in Kapitel 4.1.2.2 beschrieben. Damit die XML-Dateien für die jeweiligen SDMX-Strukturen nicht per Hand geschrieben werden müssen, werden diese automatisch durch FME-Prozesse erzeugt. Als Grundlage für die Automatisierung müssen verschiedene Konfigurationsdateien mit Inhalten befüllt werden. Für die Konfigurationsdateien wurde das Dateiformat CSV ausgewählt, da dieses sowohl automatisiert als auch manuell gut verarbeitet werden kann. Dazu ist CSV als textbasiertes Format sehr gut für Git geeignet, da Anpassungen durch Git direkt erkannt werden. OpenCoDE stellt CSV-Dateien in der Weboberfläche direkt als Tabelle dar.

Die Datei *concept_overview.csv* enthält eine Auflistung von allen Concepts mit zugehörigen Codelists. Die Auflistung ist themenübergreifend und gilt somit für die gesamte .Stat Suite. In der Tabelle (siehe Tabelle 14) werden alle Concepts mit einem technischen Namen und einer Version aufgelistet. Für die technische Repräsentation wird eine Codelist inklusive Versionsnummer für jedes Concept hinterlegt. Für die Beschreibung der Fachlichkeit werden zudem Namen und Beschreibungen des Concepts auf Deutsch und Englisch angegeben. Durch diese Definition können Codelists für verschiedene Concepts wiederverwendet werden. Innerhalb der Tabelle muss die Kombination aus Technischer Name und Version eindeutig sein. Sollte die Version einer Codelist angepasst werden, muss auch die Version des Concepts erhöht werden. Die Versionsnummern von Concept und Codelist müssen jedoch nicht identisch sein.

Tabelle 14: Auszug aus der concept_overview.csv zur Beschreibung von Concepts mit zugehörigen Codelists

Technischer Name	Name (de)	Name (en)	Beschreibung (de)	Beschreibung (en)	Codelist-Referent ID	Version
C_OBS_STATUS	Beobachtungsstatus	Observation Status			CL_OBS_STATUS(1.0)	1.0
C_SUBSTANCES	Substanzen	Substances	Liste verschiedener Substanzen	List of various substances	CL_SUBSTANCES(1.0)	1.0
C_SUBSTANCES	Substanzen	Substances	Liste verschiedener Substanzen	List of various substances	CL_SUBSTANCES(1.1)	1.1

Zu jedem Concept muss die zugehörige Codelist separat definiert werden. Die Definition erfolgt in den Unterordnern von *cube* und ist unterteilt in *metadata.csv* und *elements.csv*.

metadata.csv ist ähnlich aufgebaut wie die Konfigurationsdatei für Concepts und beinhaltet übergeordnete Informationen zu jeder Codelist. Tabelle 15 zeigt beispielhaft die Datei für die Codelist CL_SUBSTANCES. Name und Beschreibung können wie in diesem Beispiel identisch zum jeweiligen Concept sein, da eine Codelist jedoch auch von mehreren Concepts verwendet werden kann, sind hier Unterschiede möglich. Zur Erinnerung: Die Codelist beschreibt lediglich die Inhalte einer Dimension, das Concept jedoch den fachlichen Kontext. Die ID muss stets mit dem Präfix CL_ definiert werden und muss wie in der Concepts-Tabelle gemeinsam mit der Versionsnummer eindeutig für die gesamte .Stat Suite sein.

Tabelle 15: metadata.csv Tabelle der Codelist CL_SUBSTANCES

Agency ID	Version	Id	Name_de	Name_en	Beschreibung_de	Beschreibung_en
UBA	1.1	CL_SUBSTANCES	Substanz	Substances	Liste mit verschiedenen Substanzen	List of various substances

Zu jeder Codelist müssen auch die konkreten Inhalte definiert werden. Diese Definition wird in der Datei *elements.csv* durchgeführt, welche parallel zur *metadata.csv* Datei im jeweiligen Codelist Ordner abgelegt werden muss. Tabelle 16 zeigt einen Auszug der Codelist Elemente aus der entsprechenden Datei. Der Code muss innerhalb der Codelist eindeutig sein und wird später in den Daten verwendet. Generell sind Klein- und Großbuchstaben von A bis Z und Ziffern von 0 bis 9 gültige Zeichen für Codes. Zusätzlich sind als Sonderzeichen das Minus („-“), der

Unterstrich („_“) und das At-Zeichen („@“) zulässig. Da grundsätzlich Leerzeichen innerhalb eines Codes nicht erlaubt sind, können die Sonderzeichen verwendet werden, um einen Code aus mehreren Teilstücken zusammenzusetzen. Name und Beschreibung können beliebig zur Darstellung gewählt werden.

Tabelle 16: elements.csv Datei der Codelist CL_SUBSTANCES zur Definition von Codelist Elementen

Code	ParentCode	Name_de	Name_en	Beschreibung_de	Beschreibung_en
O3		Ozon	Ozone	Ozon (O3)	Ozone (O3)
NH3		Ammoniak	Ammonia	Ammoniak (NH3)	Ammonia (NH3)
NMVOC		Flüchtige organische Verbindungen ohne Methan	Non-methane volatile organic compounds	Flüchtige organische Verbindungen ohne Methan (NMVOC)	Non-methane volatile organic compounds (NMVOC)

Die Spalte ParentCode kann für die Erzeugung von Hierarchien verwendet werden. Soll eine Hierarchie aufgebaut werden, so muss der Code eines Elements (das Elternobjekt) bei einem anderen Element als ParentCode definiert werden. Diese Verschachtelungen können beliebig tief konfiguriert werden, wobei jeder Code jedoch nur das direkte Elternelement referenzieren muss. Tabelle 17 zeigt die Definition einer einfachen Hierarchie aus der Codelist CL_TRANSPORT_GOOD. Diese wird im Data-Explorer wie folgt dargestellt:

- Alle Transportgüter
  - Personentransport
  - Gütertransport.

Tabelle 17: Beispiel elements.csv zur Definition von Hierarchien (Spalten reduziert)

Code	ParentCode	Name_de
TG		Alle Transportgüter
PV	TG	Personentransport
GV	TG	Gütertransport

5.2.2.1.2 Dataflows

Analog zu den Concepts und Codelists des vorherigen Absatzes, werden auch Dataflows durch verschiedene Dateien konfiguriert.

Der Ordner *dataflows* ist unterteilt in verschiedene Themen des UBA. Diese sind zum aktuellen Zeitpunkt:

- Verkehr
- Fläche, Boden, Land-Ökosysteme
- Umwelt und Wirtschaft
- Klima
- Private Haushalte und Konsum
- Ressourcen und Abfall
- Luft
- Land- und Forstwirtschaft
- Wasser

Für jedes Thema existiert ein Ordner mit einer Konfigurationsdatei (*overview_matrix.csv*) zur Auflistung und Definition der einzelnen Dataflows. Für jeden Dataflow existiert ein weiterer Unterordner mit dem technischen Namen des jeweiligen Dataflows.

Wie bereits in Kapitel 5.2.1.2 beschrieben, wird zur Definition der Dataflows eine entsprechende Tabelle befüllt. Für jeden Dataflow wird definiert, welche Concepts benötigt werden. Dadurch kann später automatisch die SDMX-Struktur erzeugt werden. Die fachliche Beschreibung des Dataflows wird ähnlich zu Codelists in der Datei *metadata.csv* durchgeführt. Diese in dem jeweiligen Ordner des Dataflows zentral abgelegt. Tabelle 18 zeigt beispielhaft eine Dataflow-Konfiguration für DF_AIR_EMISSION_TRENDS. Die Mehrsprachigkeit wurde aus Platzgründen aus der Tabelle entfernt. Name, Beschreibung und externe Ressourcen können jedoch in Deutsch und Englisch angegeben werden. Der Technische Namen muss identisch zur Definition in der *overview_matrix.csv* angegeben sein. Durch die Spalte Kategorien kann der Dataflow einem oder mehreren Themen zugeordnet werden. Dies ist wichtig für die Darstellung und Suche im Data-Explorer. Externe Ressourcen können beliebige URLs zu weiterführenden Webseiten beinhalten, die im Data-Explorer eingebunden werden sollen. In der Spalte Filter können Standard-Konfigurationen für den Data-Explorer angegeben werden. Durch diese können Vorauswahlen der Dimensionsfilter getroffen werden, um den Datensatz bereits beim initialen Laden einzuschränken. Die genaue Definition der Filtermöglichkeiten kann in der .Stat Suite Dokumentation (Statistical Information System Collaboration Community - SIS-CC, o. J. - d) nachgelesen werden.

Tabelle 18: metadata.csv Datei des Dataflows AIR_EMISSION_TRENDS (vereinfacht)

Technischer Name	Dataflow Name (de)	Beschreibung (de)	Kategorien	Externe Ressourcen	Filter
DF_AIR_EMISSIONS_TRENDS	Emission von Luftschadstoffen	Jährliche Emissionsmengen von Luftschadstoffen, Schwermetallen und POPs	AIR, EMISSION	https://iir.umweltbundesamt.de	D_UNIT=T

Technischer Name	Dataflow Name (de)	Beschreibung (de)	Kategorien	Externe Ressourcen	Filter
		(persistente organische Schadstoffe) aus allen relevanten Quellen im Zeitverlauf von 1990 bis heute			

Zusätzlich zur Datei metadata.csv werden je Dataflow Ordner auch die Transformationen und Daten abgelegt.

Übergeordnete Transformationen zum Beispiel zur Erzeugung von SDMX-Strukturen sowie zur Automatisierung von regelmäßigen Dataflow-Aktualisierungen werden in dem Ordner *transformations* abgelegt. Die Transformationen werden im nächsten Absatz genauer beschrieben.

5.2.2.2 Erzeugung von SDMX-Strukturen

SDMX-Strukturen werden durch FME-Prozesse, sogenannten Workspaces, anhand der im vorherigen Abschnitt beschriebenen Konfigurationsdateien automatisch erzeugt. Die im Folgenden beschriebenen Workspaces sind in dem Ordner *transformations* abgelegt.

Nach der Bearbeitung der Konfigurationsdateien können alle SDMX-Dateien durch einen FME Workspaces erstellt werden. Der Prozess *sdmx_pipeline.fmw* bündelt die einzelnen Prozessschritte:

- Generierung der Codelists
- Erstellung der Dataflows

Im Folgenden werden die einzelnen Schritte erläutert.

Für die Generierung der Codelists werden in einem Steuerungsprozess (*create_sdmx_codelist_runner.fmw*) alle *metadata.csv* Dateien aus dem *cube* Ordner identifiziert. Für jede Datei muss eine Codelist erzeugt werden, für welche der Workspace *create_sdmx_codelist_from_template.fmw* mit dem entsprechenden Pfad zur Konfigurationsdatei gestartet wird. Innerhalb dieses Workspaces werden zunächst die Dateien *metadata.csv* und *elements.csv* eingelesen. Anschließend werden verschiedene Validierungsschritte durchgeführt:

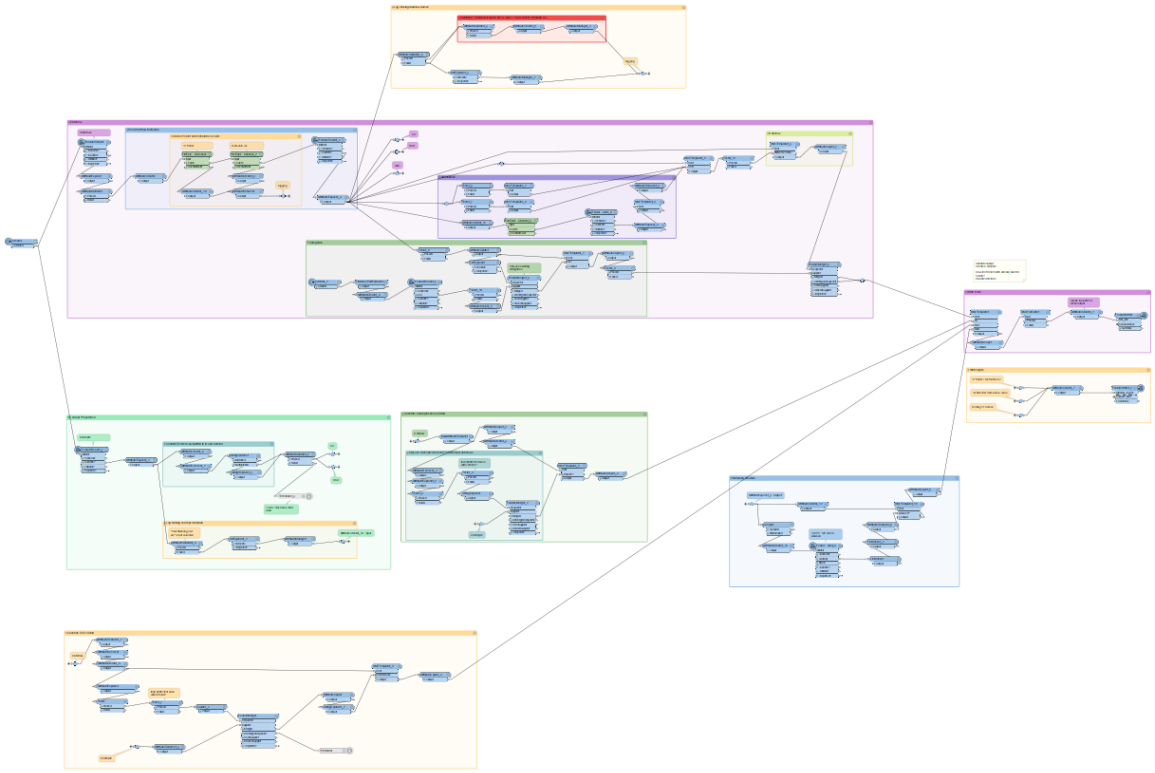
- Alle Pflichtfelder müssen gefüllt sein
  - Elemente: Code, Name (de), Name (en)
  - Metadaten: Agency ID, ID, Name (de), Name (en), Version
- In Codes dürfen keine Duplikate existieren

Nach der Validierung werden mögliche Sonderzeichen in Codes durch _ ersetzt, um technische Fehler zu vermeiden. Die eigentliche XML-Definition erfolgt durch den XMLTemplater

Transformer. In diesem können XML-Strukturen definiert werden, in dem XML-Fragmente erzeugt und mit Inhalten befüllt werden können. Pro Codelist wird eine XML-Datei erzeugt, welche sowohl die Metadaten an dem übergeordneten Codelist-Fragment als auch die Liste an konkreten Codes enthält. Das Ergebnis wird in dem jeweiligen *output* Ordner als XML-Datei mit dem Namen der Codelist abgelegt. Sollten Fehler auftreten, werden diese in einer separaten Logdatei pro Codelist protokolliert.

Zusätzlich zu den Codelists werden die Dataflow SDMX-Strukturen erzeugt. Wie in Kapitel 4.1.2.2 beschrieben, sind mehrere Strukturelemente notwendig, um einen Dataflow abzubilden (Dataflow, DataStructures, ConceptScheme, Dimensions). Zunächst werden im *sdmx_pipeline.fmw* Workspace alle *overview_matrix.csv* Dateien aus dem dataflow Ordner identifiziert. Für jede Datei wird der Workspace *create_sdmx_from_matrix.fmw* gestartet. Das bedeutet, dass pro Matrix immer alle Dataflows SDXM-Strukturen für ein gesamtes Kapitel erzeugt werden. In dem Workspace werden verschiedene Prozessschritte umgesetzt. Zunächst muss die jeweilige *overview_matrix.csv* sowie die *concept_overview.csv* eingelesen werden. Wie auch bei den Codelists, werden zunächst Pflichtfelder der Dataflow-Matrix validiert (Technischer Name, Version), anschließend wird für jeden Dataflow auch die entsprechende *metadata.csv* Datei eingelesen. Innerhalb der Dataflow SDMX-Definition werden alle Metadaten festgehalten. Dazu werden Annotations (z.B. für Filter) und Categorizations (UBA-Themen) verlinkt werden. Anhand der Matrix können alle Concepts und Codelists identifiziert werden, welche für den jeweiligen Dataflow benötigt werden. Durch diese können demnach die SDMX-Strukturen für ConceptScheme und DataStructure (inkl. Dimensionen) generiert werden. Zur Abbildung von referentiellen Metadaten wird zusätzlich eine MetadataStructure erzeugt. Diese setzt sich aus einer Liste von allgemeingültigen Metadaten und der Liste an konkreten Dimensionen des Dataflows zusammen. Die einzelnen Komponenten können wie auch bei den Codelists durch XMLTemplater erstellt werden. Da alle Komponenten spezifisch für einen Dataflow sind, können alle einzelnen XML-Fragmente in einer einzigen XML-Datei pro Dataflow zusammengefasst werden. Diese wird im *output* Ordner des Dataflows abgelegt. Auch hier werden mögliche Fehler in den Konfigurationsdateien in entsprechenden Logdateien protokolliert. Abbildung 37 zeigt einen groben Überblick über den Prozess. Die einzelnen Bookmarks (farbige Kästen) gruppieren die einzelnen SDMX-Komponenten, die rechts im Bild in einem zentralen XMLTemplater gebündelt und in einer Datei herausgeschrieben werden.

Abbildung 37: Darstellung des create_sdmx_from_matrix.fmw Workspaces zur Erzeugung von Dataflow-XML-Dateien

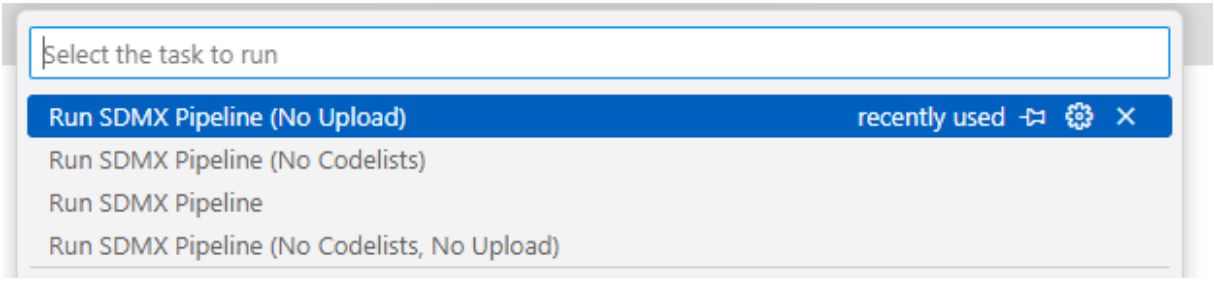


Quelle: eigene Darstellung, con terra GmbH

Zuletzt werden in dem *sdmx_pipeline.fmw* Workspace alle SDMX-Strukturen zur Fusion Metadata Registry hochgeladen, damit diese dort eingesehen werden können. Dieser Schritt wird jedoch in Zukunft entfallen, da die Fusion Metadata Registry nicht weiter betrieben wird. Der Upload in den Data Lifecycle Manager erfolgt nicht automatisch in der Pipeline, um zunächst eine manuelle Qualitätssicherung zu ermöglichen. Die XML-Dateien können anschließend manuell, oder durch den Prozess *upload_sdmx_structure_to_statsuite.fmw* hochgeladen werden.

Da viele Prozessschritte das Arbeiten mit CSV- und XML-Dateien beinhalten, wird die Arbeit durch einen guten Editor stark vereinfacht. Im Projektteam wurde hierfür Visual Studio Code (VS Code) (Microsoft, o. J.) von Microsoft verwendet. In VS Code kann das gesamte Git Verzeichnis als Projekt geöffnet werden. Alle Inhalte sind automatisch durchsuchbar und durch die Erweiterungen "Edit csv", "XML" und "Excel Viewer" können viele Schritte direkt in der Anwendung durchgeführt werden. Darüber hinaus wurden verschiedene VS Code Tasks zum Starten der SDMX-Pipeline entworfen, um die SDMX-Generierung direkt aus der Entwicklungsumgebung heraus zu starten. Abbildung 38 zeigt die verschiedenen Tasks als Auswahlliste. "Run SDMX Pipeline" startet die gesamte SDMX-Pipeline. "No Upload" sorgt dafür, dass die SDMX-Fragmente nicht in die Fusion Metadata Registry hochgeladen werden. Dies ist sinnvoll, wenn die SDMX-Fragmente noch keinen finalen Stand erreicht haben. "No Codelists" erstellt nur die Dataflow-SDMX-Fragmente neu, um die Laufzeit zu reduzieren. Die Tasks sind als Python Skript (*sdmx_pipeline.py*) implementiert. Um den Task zu nutzen, muss entweder das Verzeichnis der *fme.exe* in der PATH-Umgebungsvariable des Systems hinterlegt sein, oder das jeweilige Installationsverzeichnis muss im Python Skript in *fme_locations* hinterlegt werden. Die Tasks sind als JSON-Dateien im Ordner *.vscode* im Git hinterlegt und müssen nicht explizit installiert werden.

Abbildung 38: VS Code Task zum Starten der SDMX-Pipeline



Quelle: eigene Darstellung, con terra GmbH

Im folgenden Abschnitt wird beschrieben, wie anschließend die Daten aufbereitet werden, nachdem die SDMX-Strukturen erzeugt wurden.

5.2.2.3 Datentransformation

Der redaktionelle Prozess zum Bearbeiten von Daten wurde bereits in Kapitel 4.2 beschrieben. In diesem Abschnitt wird die technische Umsetzung der Datentransformation durch FME-Prozesse erläutert. Dabei werden sowohl einzelne Transformationen als auch automatische Aktualisierungen in regelmäßigen Intervallen betrachtet. Zum Zeitpunkt dieses Berichts wurden nach dem beschriebenen Vorgehen bereits 57 Dataflows in der .Stat Suite bereitgestellt.

5.2.2.3.1 Transformation von Daten nach SDMX-CSV

Um Daten in die .Stat Suite importieren zu können, müssen diese in das SDMX-CSV Format überführt werden. Dieser Schritt wurde bereits in Kapitel 4.2.1.2.2 für den allgemeinen redaktionellen Prozess beschrieben. Insgesamt ist die Komplexität sehr stark von dem jeweiligen Quelldatensatz abhängig. Je ähnlicher ein Datensatz bereits der Zielstruktur (SDMX-CSV) ist, desto geringer ist der Aufwand. Trotzdem ähneln sich viele Aufgaben, die innerhalb der Datentransformation durchgeführt werden müssen. In diesem Abschnitt werden zunächst allgemeine Transformationsschritte erläutert, bevor diese dann an einem konkreten FME Workspace erläutert werden.

5.2.2.3.1.1 Allgemeine Schritte zur Datentransformation

Zunächst werden die Quelldaten eingelesen. Zum aktuellen Zeitpunkt werden Daten aus CSV-, Excel-, XML-Formaten und verschiedenen REST-Endpunkten (DWD, Destatis, UBA) importiert. FME unterstützt bereits nativ eine Vielzahl von verschiedenen Formaten, die direkt eingelesen werden können, trotzdem ist eine Aufbereitung der Daten unumgänglich. Im Beispiel von Excel müssen unter anderem die korrekten Tabellenblätter und die relevanten Zellen definiert werden. Einige Excel-Dateien beinhalten mehrere Tabellen pro Tabellenblatt, die nur durch eine manuelle Zuordnung fachlich unterschieden werden können. Dies bedeutet, dass die jeweiligen Reader in den Workspaces entsprechend konfiguriert werden müssen. Bei APIs hingegen muss die entsprechende Aufruf-URL definiert werden. Das Ergebnis der meisten APIs ist entweder ein JSON- oder XML-Fragment, welches anschließend auszuwerten ist.

Darauffolgend werden die Quelldaten bei Bedarf gefiltert, wobei einzelne Zeilen oder Spalten zum Beispiel aus der Quelltabelle entfernt werden können. Gründe hierfür können u.a. Datenschutz oder fehlerhafte Daten sein.

In einigen Fällen müssen Daten zunächst transponiert werden. Tabelle 11 in Kapitel 4.2.1.4.1 zeigt ein Beispiel für einen Quelldatensatz, in dem Messwerte der einzelnen Jahre in Spalten angegeben werden. Die Tabelle "wächst" also für weitere Jahre in die Breite. Für SDMX-CSV

müssen die Spalten jedoch die einzelnen Dimensionen beinhalten und die für weitere Messwerte müssen weitere Zeilen eingefügt werden.

Der aufwendigste Prozessschritt ist häufig das Mapping von Dimensionswerten auf die vereinheitlichten SDMX-Codelists. Wie in Kapitel 5.2.1 erläutert, werden Codelists wenn möglich themenübergreifend vereinheitlicht um eine homogene Darstellung der Daten innerhalb der .Stat Suite zu ermöglichen. Dies bedeutet, dass die Quelldaten neuen Codes zugewiesen werden müssen, damit die entsprechende Dimension über die jeweilige Codelist korrekt dargestellt werden kann. Hierzu werden in der Regel Mappingtabellen angelegt. Ein einfaches Beispiel ist die Angabe des Bundeslandes. In den Daten könnte z.B. "Nordrhein-Westfalen" stehen, während der Code nur "NRW" lautet.

Einige Dimensionen beinhalten Informationen, die für den gesamten Dataflow gesetzt werden müssen, die jedoch nicht in den Daten enthalten sind. Diese müssen im Prozess ebenfalls an alle oder ausgewählte Elemente geschrieben werden. Beispiele hierfür sind die Frequenz der Daten, Kommentare und Fußnoten.

Je nach Datensatz werden zusätzlich Berechnungen in den Daten durchgeführt. Einfache Fälle sind Konvertierungen von Einheiten zur Vereinheitlichung von Datensätzen. In einigen Datensätzen ist jedoch auch die Berechnung von statistischen Kennwerten oder Regressionen notwendig.

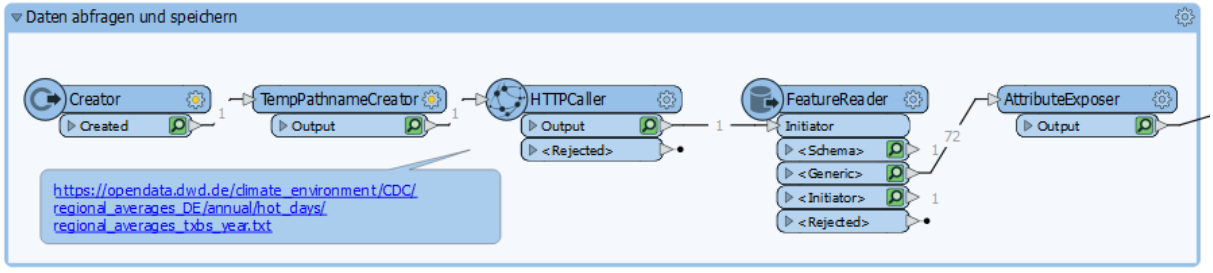
Zuletzt müssen die Dimensionen entsprechend der Dataflow-Definition benannt werden und der Datensatz kann als CSV-Datei durch den nativen CSV-Writer gespeichert werden. Sobald der Prozess einmal definiert ist, kann dieser beliebig häufig ausgeführt werden. Dies ist insbesondere wichtig, um effiziente Datenaktualisierungen vorzunehmen. Solange die Datenstruktur unverändert bleibt, kann der Datensatz mit dem gleichen Prozess aktualisiert werden.

5.2.2.3.1.2 Beispiel FME Prozess zur Datenintegration

Im Folgenden wird ein FME-Prozess zur Erstellung von SDMX-CSV Dateien am Beispiel des Dataflows "Germany Hot Days" erläutert.

Der originale Datensatz wird durch den DWD bereitgestellt und kann als CSV-Datei über die URL https://opendata.dwd.de/climate_environment/CDC/regional_averages_DE/annual/hot_days/regional_averages_txbs_year.txt heruntergeladen werden. In dem Datensatz wird die Anzahl der Tage mit einem Lufttemperatur-Maximum von mehr als 30 Grad Celsius pro Bundesland aufgeschlüsselt. Im ersten Schritt (Abbildung 39) werden die Daten in FME über den HTTPCaller heruntergeladen und durch einen FeatureReader als CSV-Datei ausgelesen. Durch den TempPathnameCreator wird eine temporäre Datei angelegt, die automatisch nach Prozessende gelöscht wird.

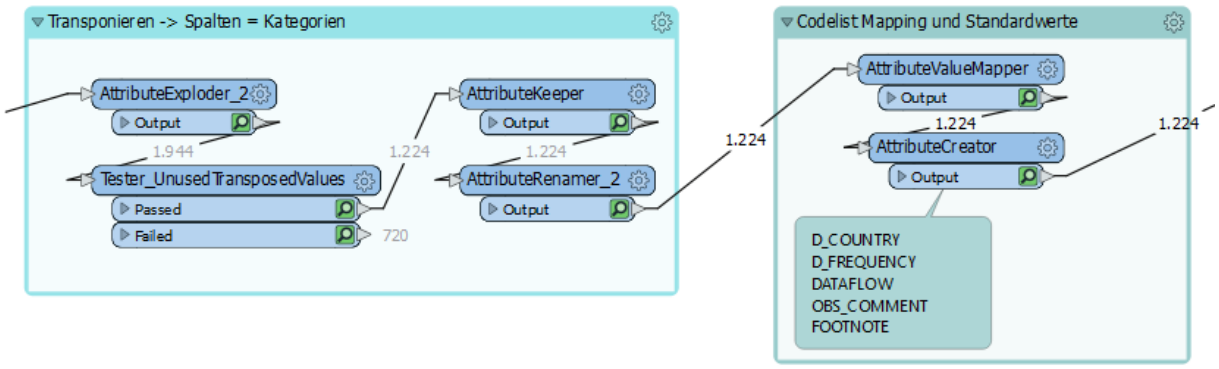
Abbildung 39: Daten des DWD durch FME einlesen



Quelle: eigene Darstellung, con terra GmbH

Die CSV-Datei beinhaltet eine Spalte für das aktuelle Jahr und jeweils eine Spalte pro Bundesland. Wie zuvor erläutert muss diese Datenstruktur jedoch umstrukturiert werden, so dass jeder Messwert in einer Zeile steht und alle dazugehörigen Dimensionen in den Spalten aufgeführt werden. Dieser Schritt (Abbildung 40) wird durch einen AttributeExploder ermöglicht. Dieser Transformer erzeugt ein neues Feature (ein einzelnes Objekt in FME) für jedes Attribut am bisherigen Feature. Dadurch wird von einer Spalten-Anordnung der Daten zu einer Zeilen-Anordnung gewechselt. Nicht benötigte Inhalte können durch einen Tester entfernt werden. Anschließend werden verschiedene Attribute umbenannt, um den Dataflow-Dimensionen zu entsprechen (zum Beispiel "Jahr" zu "TIME_PERIOD"). Anschließend werden die Bundesländer zu den entsprechenden Codes im AttributeValueMapper zugeordnet (zum Beispiel "Baden-Wuerttemberg" zu "BW"). Statische Informationen wie die Frequenz der Daten, das Land oder auch der Dataflowname können als konstante Werte in einem AttributeCreator erzeugt werden.

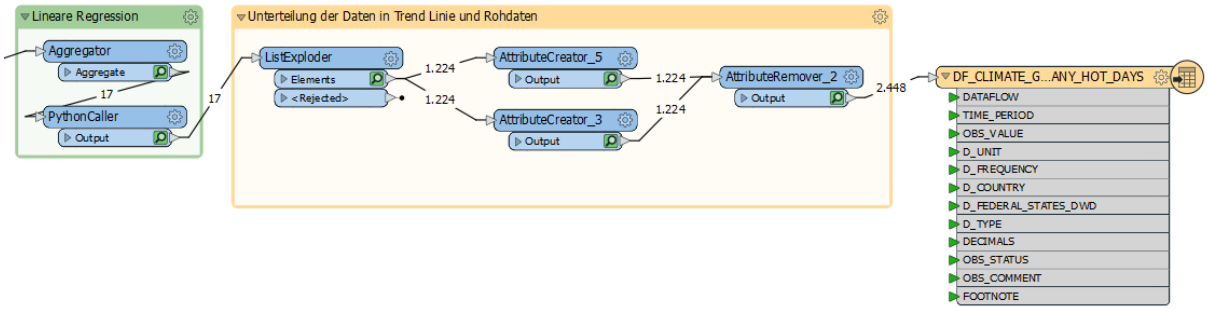
Abbildung 40: Transponieren der CSV-Daten und setzen von Dimensionswerten



Quelle: eigene Darstellung, con terra GmbH

Nach den bisherigen Schritten könnte der Datensatz bereits als SDMX-CSV geschrieben werden. Für den konkreten Dataflow war jedoch zusätzlich zu der reinen Darstellung der Messwerte auch eine Trendlinie zu berechnen. In Abbildung 41 werden hierfür zunächst alle Daten durch einen Aggregator zu einer Liste zusammengefasst. FME ermöglicht das Einbetten von Python-Code. In dem Transformer PythonCaller wird den Trendlinie durch eine entsprechende numpy-Funktion berechnet und ebenfalls als Liste an dem jeweiligen Feature angehängt. Die zuvor aggregierte Liste wird nun wieder aufgelöst (ListExploder). Zur Unterscheidung zwischen linearem Trend und Jahreswerte wird durch einen AttributeCreator jeweils der entsprechende Code des Datentyps ergänzt. Für die Trendlinie wird an jedem Messwert noch der OBS_STATUS "Eigene Berechnung" gesetzt. Zuletzt können die Daten durch einen CSV-Writer dateibasiert gespeichert werden.

Abbildung 41: Berechnung einer Trendlinie und Schreiben der Daten als CSV



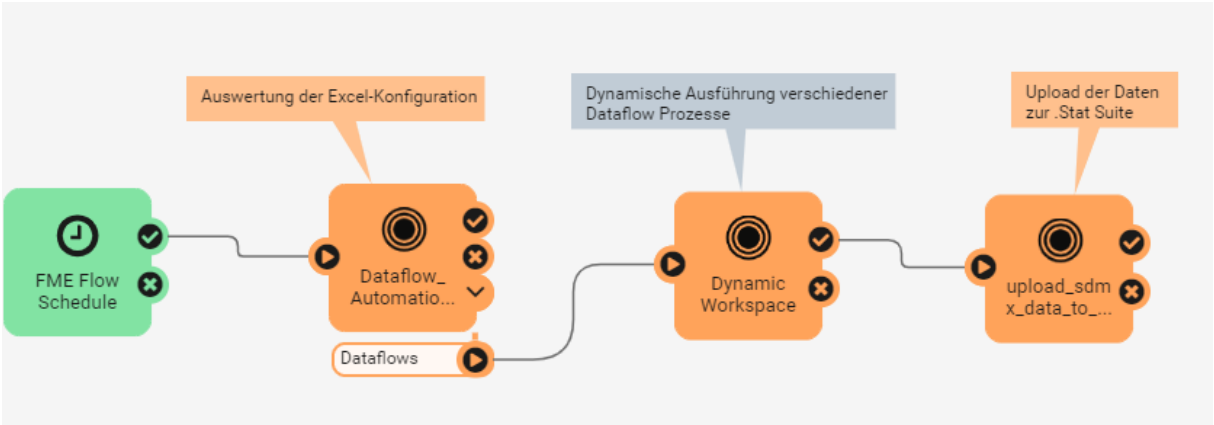
Quelle: eigene Darstellung, con terra GmbH

5.2.2.3.2 Automatisierte Dataflow Aktualisierungen

Viele Dataflows beinhalten Zeitreihen, die fortlaufend aktualisiert werden, wobei das Intervall jeweils unterschiedlich sein kann. Als Grundvoraussetzung für eine automatische Aktualisierung gilt jedoch stets, dass die Datenquelle über entsprechende Schnittstelle abgerufen werden kann und dass die Datenstruktur unverändert bleibt. Sind diese Bedingungen erfüllt, so können Dataflows über den im Folgenden beschriebenen Prozess aktualisiert werden. Der Workflow zum Updaten von Dataflows in der .Stat Suite ist in Kapitel 4.2.1.3 beschrieben.

Für die automatische Aktualisierung der Dataflows wird der FME Server des UBA verwendet. Die Installation wird zentral verwaltet und ist nicht Teil des Projektes. Auf eine Beschreibung der Infrastruktur wird an dieser Stelle daher verzichtet. Innerhalb des FME Server wird hierzu eine sogenannte Automation erstellt, durch welche Prozessketten in regelmäßigen Intervallen gestartet werden können. Abbildung 42 zeigt die verwendete Automation. Sie beginnt mit einem Schedule-Trigger, welcher die Automation jede Nacht um 02:00 Uhr startet. Nach dem Start wird eine Konfigurationsdatei (siehe Tabelle 19) ausgewertet, in der alle zu aktualisierende Dataflows aufgelistet sind. Ist ein Dataflow bereit zur Aktualisierung, so wird der jeweilige FME-Prozess ausgeführt, um die SDMX-CSV Datei zu erzeugen. Diese wird im letzten Prozessschritt zur .Stat Suite hochgeladen.

Abbildung 42: FME Server Automation zum automatischen Updaten von Dataflows



Quelle: eigene Darstellung, con terra GmbH

Für die Automatisierung müssen zunächst die Workspaces für alle zu automatisierenden Dataflows auf dem FME Server publiziert werden. Hierzu können die in Kapitel 5.2.2.3.1.2 beschriebenen Prozesse verwendet werden. Für die Automatisierung ist jedoch wichtig, dass der Zielordner für die SDMX-CSV Datei als Prozessparameter definiert wird, damit dieser der

Automatisierung zur Verfügung steht. Konkret bedeutet dies, dass die CSV-Dateien auf dem FME Server nicht in das Git, sondern in ein temporäres Verzeichnis geschrieben werden, welches nur für den Upload verwendet wird. Um eine generische Ausführung der Prozesse zu ermöglichen, muss dieser Parameter bei allen Workspaces identisch benannt werden (SDMX_CSV_OUTPUT). Dazu muss der Workspace mit einem eindeutigen Namen auf dem Server publiziert werden.

In der Konfigurationstabelle (Tabelle 19) wird definiert, wann jeder Dataflow aktualisiert werden soll. Hierzu wird jeder Dataflow mit der eindeutigen ID und Informationen zum Intervall aufgelistet. Die Dataflow ID muss an dieser Stelle identisch mit dem publizierten Workspace-Namen sein. Gültige Zeitintervalle sind *daily*, *weekly*, *monthly* und *yearly*, die jeweils in der Spalte Interval einzutragen sind. Standardmäßig werden die nicht-täglichen Intervalle am jeweils ersten Tag der Periode (z.B. Monatserster) ausgeführt. In der Spalte Interval_Details können optional mehrere Ausführungen kommagetrennt und unter Beachtung der jeweiligen Wertelimits definiert werden:

- weekly: 1-7 (Montag bis Sonntag)
- monthly: 1-31
- yearly: 1-12

Beispiel: Interval=weekly; Interval_Details=2,5 bedeutet eine wöchentliche Ausführung jeden Dienstag und Freitag. Interval=yearly; Interval_Details=5 bedeutet eine jährliche Ausführung am 01.05.

Tabelle 19: Tabelle zur Konfiguration von Dataflow-Automatisierungen (Beispielkonfiguration)

Dataflow_ID	Interval	Interval_Details
DATAFLOW_1	daily	
DATAFLOW_2	weekly	2,5
DATAFLOW_3	monthly	
DATAFLOW_5	yearly	5

Der Upload der SDMX-CSV Dateien erfolgt über die REST-Schnittstelle des .Stat Suite Transfer-Services (Statistical Information System Collaboration Community - SIS-CC, o. J. - aj). Hierzu wird zunächst die CSV-Datei über den Endpunkt für SDMX-Importe (**Fehler! Linkreferenz ungültig.**) hochgeladen. Da es sich um eine asynchrone Operation handelt wird als Rückgabe nur die Transfer-ID bereitgestellt. Diese kann zur Überprüfung des Transfer-Status (**Fehler! Linkreferenz ungültig.** Service URL> /2/status/request) verwendet werden, welcher so oft ausgeführt wird, bis der Transfer entweder als Erfolg oder Fehlschlag gekennzeichnet wird. Um unnötige Uploads und damit eine Prozessauslastung der .Stat Suite zu verhindern, werden Datensätze nur hochgeladen, falls eine Veränderung vorliegt. Hierzu wird vor dem Upload ein File-Hash berechnet und mit Zeitstempel in einer lokalen Datenbank abgelegt. Dateien werden nur hochgeladen, wenn sich der File-Hash von dem vorherigen Upload unterscheidet. Nach einem erfolgreichen Upload wird der Hash in der Datenbank für den jeweiligen Dataflow aktualisiert.

5.2.2.4 Metadaten

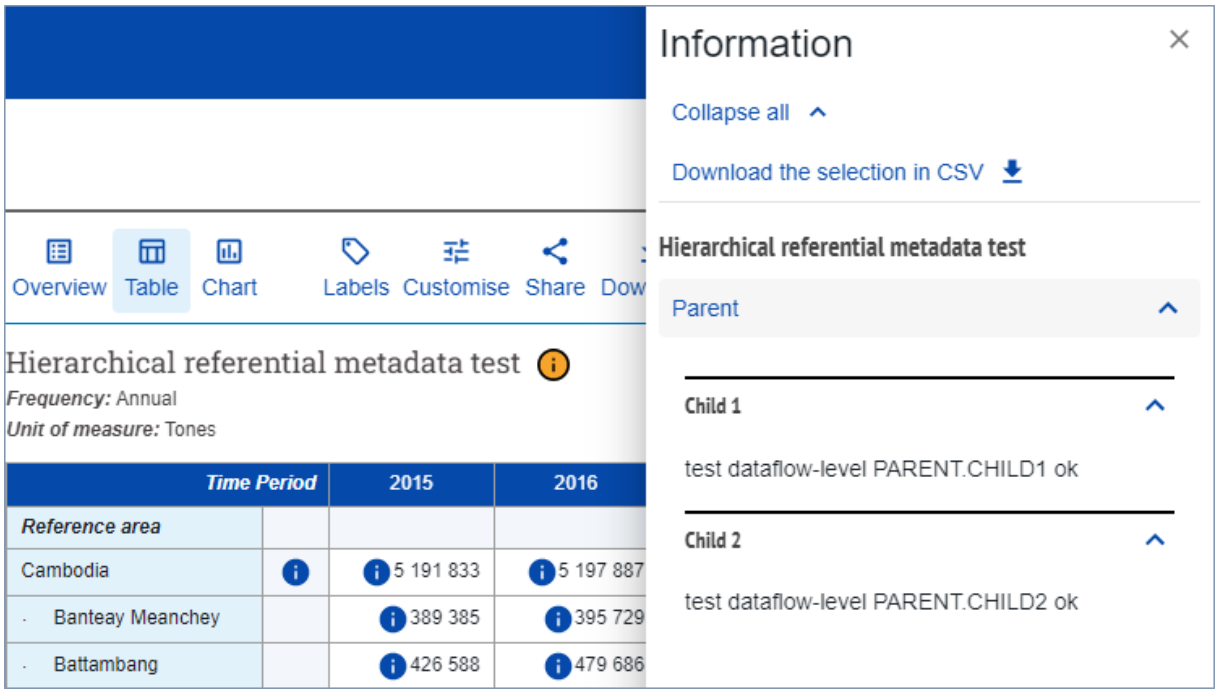
Zu allen Datensätzen, die im Data Cube publiziert werden sollen, existieren Metadaten, die ebenfalls gespeichert und dargestellt werden müssen. Beispiel sind Titel, Beschreibung, aber auch weitere Verlinkungen, Nutzungsbedingungen und viele weitere. In diesem Kapitel wird zunächst auf Möglichkeiten der .Stat Suite eingegangen, Metadaten zu importieren und anzuzeigen. Anschließend wird die Verknüpfung mit dem UBA-Metadatenkatalog zur automatischen Synchronisierung der Metadaten aus einem zentralen Datenbestand beschrieben.

5.2.2.4.1 Metadaten in der .Stat Suite

Auf der Startseite eines Dataflows im Data-Explorer werden einfache Metadaten direkt dargestellt. Die Darstellung beinhaltet den Titel, die Beschreibung, Anzahl der Datenpunkte, Aktualisierungsdatum, die Dimensionen und mögliche Verlinkungen zu verwandten Dateien/ Webseiten. Die entsprechenden Informationen werden hierzu direkt im SDMX-XML des jeweiligen Dataflows hinterlegt.

Weitere Metadaten werden im "Information Side Panel" (Statistical Information System Collaboration Community - SIS-CC, o. J. - ao) der Tabellenansicht dargestellt (siehe Abbildung 43). Metadaten können an verschiedenen Ebenen des Dataflows referenziert werden, wobei diese als "i" Symbol in der Tabelle oder an der Überschrift gekennzeichnet werden. Durch einen Klick auf das Symbol öffnet sich das Information Side Panel. Metadaten können für den gesamten Dataflow, pro Tabellenzelle, oder an ganzen Spalten/ Zeilen hinterlegt werden. Je nachdem an welcher Stelle das Side Panel geöffnet wird, werden daher unterschiedliche Metadaten angezeigt. Wie in der Abbildung dargestellt, ist eine Gruppierung der Metadaten zur besseren Organisation möglich. Darüber hinaus können die Metadaten separat zu den eigentlichen Daten als CSV-Datei heruntergeladen werden.

Abbildung 43: Information Side Panel des Data Explorers



Quelle: (Statistical Information System Collaboration Community - SIS-CC, o. J. - ao)

Metadaten pro Zelle, also pro Messwert (OBS_VALUE), können direkt in der SDMX-CSV als Kommentar (OBS_COMMENT) hinterlegt werden. Für alle weiteren Metadaten existieren die sogenannten "Referentiellen Metadaten". Diese werden, wie der Name impliziert, separat zum

Dataflow gepflegt und entsprechend referenziert. Dies ermöglicht eine bessere Wartbarkeit, da Metadaten unabhängig von den Daten editierbar sind.

Zur Nutzung der Referentiellen Metadaten wird analog zur DataStructure nun eine MetadataStructure als SDMX-Struktur definiert. Die MetadataStructure definiert die Darstellung der Metadaten im Sidepanel, wobei die einzelnen Metadaten-Attribute hierarchisch eingeordnet und mehrsprachig definiert werden können.

Analog zur Verwendung der SDMX-CSV Dateien werden Metadaten ebenfalls in einer CSV-Datei gepflegt und über den Data Lifecycle Manager hochgeladen. Eine Beispiel-Struktur ist in Tabelle 20 dargestellt. Innerhalb der CSV-Struktur müssen alle Metadaten-Attribute als Spalten aufgeführt werden. Zusätzlich ist der entsprechende Dataflow mit Versionsnummer zu referenzieren. Je nach Anwendungsfall können Metadaten-Attribute für den gesamten Dataflow (siehe erste Zeile) oder aber auch für einzelne Dimensionen, oder Kombinationen aus Dimensionen definiert werden (siehe Zeilen zwei und drei). Eigene Metadaten (z.B. "CONTACT") müssen zuvor in der MetadataStructure definiert sein.

Tabelle 20: Beispiel SDMX-CSV für Referentielle Metadaten

STRUCTURE	STRUCTURE_ID	TIME_PERIOD	D_COUNTRY	REF_COMMENT	CONTACT
DATAFLOW	UBA:DF_Example(1.0)			Kommentar für den gesamten Dataflow	Kontaktperson
DATAFLOW	UBA:DF_Example(1.0)	2022		Kommentar nur für Daten mit Jahr 2022	
DATAFLOW	UBA:DF_Example(1.0)	2022	DE	Kommentar nur für Daten aus Deutschland mit Jahr 2022	

5.2.2.4.2 Integration in die bestehenden SDMX-Prozesse

Für eine einheitliche Darstellung der Metadaten über alle Dataflows wurde eine Liste von Metadaten abgestimmt. Diese wird im nächsten Abschnitt 5.2.2.4.3 genauer erläutert.

Die zuvor beschriebene MetadataStructure muss für jeden Dataflow als SDMX-XML definiert und mit dem entsprechenden Dataflow verknüpft werden. Dieser Vorgang wurde in den bestehenden FME- Prozess zur Erzeugung der SDMX-Strukturen integriert (siehe Kapitel 5.2.2.2). Dies bedeutet, dass für jeden Dataflow eine eigene MetadataStructure definiert wird, die jedoch die gleichen Metadaten-Attribute beinhaltet. Zur Vereinfachung werden die Benamungen und mehrsprachige Definition der Attribute in einem zentralen ConceptScheme definiert.

Nach dem üblichen Prozess zur Erstellung der Dataflow SDMX-Strukturen können daher automatisch auch Referentielle Metadaten für den jeweiligen Datensatz importiert werden.

5.2.2.4.3 Verknüpfung mit dem UBA-Metadatenkatalog

Zum Zeitpunkt dieses Berichts wird parallel ein weiteres Projekt innerhalb des UBA mit dem Titel "Datennutzungskonzept und Dateninfrastruktur für sozial- und naturwissenschaftliche Datenbestände des Umweltbundesamtes" (Forschungskennzahl 3719 12 1070) durchgeführt. Ziel dieses Projektes ist es, ein Konzept zum Umgang mit Metadaten als organisatorischen Rahmen zu entwerfen, um einen Metadatenkatalog aufzubauen. Der Metadatenkatalog soll beschreibende Informationen zu allen Datensätzen des UBA aufnehmen, um diese durchsuchbar zu machen und strukturiert darzustellen. In diesem Absatz wird die Verknüpfung der beiden Projekte erläutert.

Zum aktuellen Zeitpunkt befindet sich der Metadatenkatalog noch in der Umsetzungsphase, die notwendige Schnittstelle zur Verknüpfung der beiden Projekte wurde jedoch bereits definiert. Der Metadatenkatalog wird eine API bereitstellen, über die alle Metadaten als DCAT-AP.de (Geschäfts- und Koordinierungsstelle GovData - GKSt, 2022) konforme XML-Dateien abgerufen werden können. DCAT-AP.de ist ein deutschlandweiter Metadatenstandard, der auf dem europäischen Standard DCAT-AP ("Data Catalogue Application Profile") aufbaut, welcher primär zur Beschreibung von Metadaten aus Datenportalen definiert wurde. DCAT-AP.de stellt verschiedene Klassen zur Verfügung, wobei für den Data Cube nur die Klassen "Datensatz" ("Dataset") und "Distribution" ("Distribution") relevant sind. Die Klasse "Datensatz" stellt beschreibende Attribute zur Verfügung, die sich direkt auf die konkreten Daten beziehen. "Distribution" beschreibt die Art, wie der genannte Datensatz zur Verfügung gestellt wird (zum Beispiel Datenformat, URL, und weitere). Eine genaue Auflistung der verwendeten Metadatenattribute folgt weiter unten. Ein Datensatz kann laut dem Standard beliebig viele Distributionen besitzen.

Für den Data Cube wurde die Entscheidung getroffen, dass es für jeden Dataflow aus der .Stat Suite genau einen Eintrag (ein "Datensatz") im Metadatenkatalog geben wird. Da die Datenbereitstellung durch die .Stat Suite fest definiert ist, ist dazu für jeden Dataflow auch nur eine Distribution notwendig.

Der Metadatenkatalog wird die zentrale Komponente zur Pflege und Sammlung aller Metadaten. Daher wurde weiter entschieden, dass Informationen vom Metadatenkatalog zur .Stat Suite synchronisiert werden sollen, nicht aber von der .Stat Suite zum Metadatenkatalog. Dadurch werden Konflikte vermieden, die entstehen würden, wenn der gleiche Datensatz an verschiedenen Stellen mit unterschiedlichen Inhalten gepflegt werden würde.

Durch DCAT-AP.de wird eine Liste von Metadatenattributen inklusive ihrer jeweiligen Kardinalität für die jeweiligen Klassen vorgegeben. Im Rahmen des Datennutzungskonzeptes wurde die Liste für das UBA noch einmal überarbeitet, um einige Attribute über den Standard hinaus als Pflichtfelder zu definieren. Die folgenden Tabellen (Tabelle 21, Tabelle 22) listen alle Metadaten auf. Dabei wird die jeweilige Quelle aus dem DCAT-AP.de einem entsprechenden Attribut der Referentiellen Metadaten (SDMX-CSV) zugeordnet. Aufgrund der Vielzahl an Attributen wurde zur vereinfachten Darstellung innerhalb der .Stat Suite zusätzlich eine Gruppierung der Attribute vorgenommen. Die folgenden Gruppen wurden definiert und sind ebenfalls in den folgenden Tabellen aufgeführt:

- Allgemeine Metadaten:
- Klassifikation und Kategorisierung:
- Qualität und Konformität:
- Beziehungen und Referenzen:

- Geografische und zeitliche Informationen:
- Distribution und Zugangsinformationen:
- Versionskontrolle und Historie:

Alle Attribute aus dem DCAT-AP.de Datensatz sollen zur .Stat Suite synchronisiert werden. Da nur eine DCAT-AP.de Distribution je Datensatz in der .Stat Suite existiert, wird nicht die gesamte Liste an Attributen benötigt, da es an vielen Stellen zu Redundanzen in der Anzeige kommt. In der Auflistung der Distributionsattributen wird daher vermerkt, ob ein Attribut zur .Stat Suite synchronisiert wird. Falls keine Synchronisation notwendig ist, so ist der jeweilige Grund in der Tabelle als Kommentar notiert.

Tabelle 21: Dataset: Metadaten Mapping DCAT-AP.de zur .Stat Suite

Name	Gruppierung	DCAT-AP.de	SDMX-CSV
Aktualisierungsdatum	Allgemeine Metadaten	dct:modified	DATASET_MODIFIEDDATE
Aktualisierungsfrequenz	Allgemeine Metadaten	dct:accrualPeriodicity	UPDATEFREQUENCY
Andere ID	Allgemeine Metadaten	adms:identifier	FURTHERID
Autor	Allgemeine Metadaten	dct:creator	CREATOR
Bearbeiter	Allgemeine Metadaten	dct:contributor	CONTRIBUTOR
Beispieldistribution	Distribution und Zugangsinformationen	adms:sample	SAMPLEDISTRIBUTION
Beschreibung	Allgemeine Metadaten	dct:description	DATASET_DESCRIPTION
Beschreibung der Abdeckung	Geografische und zeitliche Informationen	dcatde:geocodingDescription	GEOCODINGDESCRIPTION
Datenbereitsteller ID	Distribution und Zugangsinformationen	dcatde:contributorID	CONTRIBUTORID
Distribution	Distribution und Zugangsinformationen	dcat:distribution	DISTRIBUTION
Dokumentation	Allgemeine Metadaten	foaf:page	DATASET_DOCUMENTATION
Ebene der geopolitischen Abdeckung URI	Geografische und zeitliche Informationen	dcatde:politicalGeocodingLevelURI	POLITICALGEOCODINGLEVELURI
Geopolitische Abdeckung URI	Geografische und zeitliche Informationen	dactde:politicalGeocodingURI	POLITICALGEOCODINGURI
Grad der Zugänglichkeit	Qualität und Konformität	dct:accessRights	DATASET_ACCESSRIGHTS

Name	Gruppierung	DCAT-AP.de	SDMX-CSV
Herausgeber	Allgemeine Metadaten	dct:publisher	PUBLISHER
ID	Beziehungen und Referenzen	dct:identifier	IDENTIFIER
Ist Version von	Beziehungen und Referenzen	dct:isVersionOf	ISVERSIONOF
Kategorie	Klassifikation und Kategorisierung	dcat:theme	THEME
Konform zu Standard	Qualität und Konformität	dct:conformsTo	DATASET_CONFORMSTO
Kontakt	Distribution und Zugangsinformationen	dcat:contactPoint	CONTACT
Provenienz	Allgemeine Metadaten	dct:provenance	PROVENANCE
Qualifizierte Beziehung	Beziehungen und Referenzen	dcat:qualifiedRelation	QUALIFIEDRELATION
Qualitätssicherungsprozess	Qualität und Konformität	dcatde:qualityProcessURI	QUALITYPROCESS
Quelle der Datenstruktur	Allgemeine Metadaten	dct:source	SOURCEOFDATASTRUCTURE
Räumliche Abdeckung	Geografische und zeitliche Informationen	dct:spatial	SPATIALCOVERAGE
Räumliche Auflösung in Meter	Geografische und zeitliche Informationen	dcat:spatialResolutionInMeters	SPATIALRESOLUTIONINMETERS
Rechtsgrundlage für Zugangseröffnung	Qualität und Konformität	dcatde:legalBasis	LEGALBASIS
Referenzieller Kommentar	Referenzieller Kommentar		REF_COMMENT
Referenziert	Beziehungen und Referenzen	dct:references	REFERENCES
Rollenzuordnung	Beziehungen und Referenzen	prov:qualifiedAttribution	ROLEASSIGNMENT
Schlagwort	Klassifikation und Kategorisierung	dcat:keyword	KEYWORD
Sprache	Allgemeine Metadaten	dct:language	DATASET_LANGUAGE
Titel	Allgemeine Metadaten	dct:title	DATASET_TITLE
Typ des Datensatzes	Klassifikation und Kategorisierung	dct:type	TYPEOFDATASET

Name	Gruppierung	DCAT-AP.de	SDMX-CSV
Urheber	Allgemeine Metadaten	dcatde:originator	ORIGINATOR
Ursprüngliche Website	Beziehungen und Referenzen	dcat:landingPage	ORIGINALWEBSITE
Verfügbarkeit	Qualität und Konformität	dcatap:availability	DATASET_AVAILABILITY
Veröffentlichungsdatum	Allgemeine Metadaten	dct:issued	DATASET_DATEOFISSUE
Versionsbezeichnung	Versionskontrolle und Historie	owl:versionInfo	VERSIONNAME
Versionserläuterung	Versionskontrolle und Historie	adms:versionNotes	VERSIONDESCRIPTION
Verwalter	Allgemeine Metadaten	dcatde:maintainer	MAINTAINER
Verwandte Ressource	Beziehungen und Referenzen	dct:relation	RELATEDRESOURCE
Weitere Version	Beziehungen und Referenzen	dcat:DatasetSeries	FURTHERVERSION
Wird Referenziert	Beziehungen und Referenzen	dct:isReferencedBy	ISREFERENCED
Wurde erzeugt von	Beziehungen und Referenzen	prov:wasGeneratedBy	GENERATEDBY
Zeitliche Abdeckung	Geografische und zeitliche Informationen	dct:temporal	TEMPORALCOVERAGE
Zeitliche Auflösung	Geografische und zeitliche Informationen	dcat:temporalResolution	TEMPORALRESOLUTION

Tabelle 22: Distribution: Metadaten Mapping DCAT-AP.de zur .Stat Suite

Name	Gruppierung	DCAT-AP.de	SDMX-CSV	Übernahme .statSuite	Bemerkung
Zugangs-URL		dcat:accessURL	ACCESSURL		URL wäre der Data-Explorer und für Nutzende somit eine Verlinkung auf die aktuelle Seite.

Name	Gruppierung	DCAT-AP.de	SDMX-CSV	Übernahme .statSuite	Bemerkung
Lizenz	Distribution und Zugangsinformationen	dct:license	LICENSE	x	
Format		dct:format	FORMAT		In .Stat Suite immer gleich (CSV, Excel) und über die GUI bereits ersichtlich
Beschreibung		dct:description	DISTR_DESCRIPTION		Identisch mit Dataset Beschreibung
Größe in Bytes		dcat:byteSize	BYTESIZE		Größe ändert sich bei jedem Daten-Update. Es existieren keine vordefinierten Dateien..
Prüfsumme		spdx:checksum	CHECKSUM		Prüfsumme ändert sich bei jedem Daten-Update. Es existieren keine vordefinierten Dateien.
Dokumentation		foaf:page	DISTR_DOCUMENTATION	x	
Download-URL		dcat:downloadURL	DOWNLOADURL		Download wird über die Data-Explorer GUI schon abgebildet. URL wäre eine Dopplung in der GUI.
Sprache		dct:language	DISTR_LANGUAGE		Dopplung zum Dataset.
Konform zu Standard		dct:conformsTo	DISTR_CONFORMSTO		Dopplung zum Dataset.
Medientyp		dcat:mediaType	MEDIATYPE	x	

Name	Gruppierung	DCAT-AP.de	SDMX-CSV	Übernahme .statSuite	Bemerkung
Veröffentlichungsdatum		dct:issued	DISTR_DATEOFISSUE		Dopplung zum Dataset.
Grad der Zugänglichkeit		dct:rights	DISTR_ACCESSRIGHTS		Dopplung zum Dataset.
Status	Allgemeine Metadaten	adms:status	STATUS	x	
Titel		dct:title	DISTR_TITLE		Dopplung zum Dataset.
Aktualisierungsdatum		dct:modified	DISTR_MODIFIEDDATE		Bereits nativ über die .Stat Suite abgedeckt.
Namensnennungstext für "By"-Clauses	Distribution und Zugangsinformationen	dcatde:licenseAttributionByText		x	
Verfügbarkeit	Distribution und Zugangsinformationen	dcatap:availability	DISTR_AVAILABILITY	x	
Kompressionsformat		dcat:compressFormat	COMPRESSFORMAT		Angabe für alle Dataflows identisch. Keine Kompression vorhanden.
Paketformat		dcat:packageFormat	PACKAGEFORMAT		Angabe für alle Dataflows identisch. Keine Paketformate vorhanden.
Regelwerk	Distribution und Zugangsinformationen	odri:hasPolicy	POLICY	x	
Ausliefernder Datenservice	Distribution und Zugangsinformationen	dcat:accessService	ACCESSSERVICE		Verlinkung auf die aktuelle Webseite wäre redundant.

Für die Zuordnung von Metadatenkatalog zur .Stat Suite muss in dem Datensatz-Attribut "ID" die Dataflow-ID aus der .Stat Suite angegeben werden. Die Übernahme der Metadaten vom Metadatenkatalog wird anschließend regelmäßig von einem FME-Prozess durchgeführt, sobald der Metadatenkatalog fertig gestellt ist.

Der Prozess wurde bereits wie folgt umgesetzt:

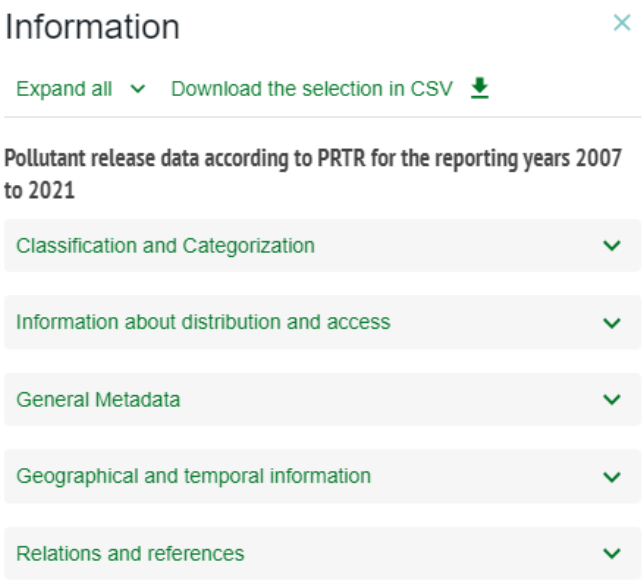
Innerhalb des FME-Prozesses wird eine XML-Datei eines DCAT-AP.de Catalogs eingelesen. Dieser beinhaltet alle Metadaten unterteilt in die einzelnen Datensätze. Nach Überprüfung der ID wird ein zugehöriger Dataflow in der .Stat Suite für das Mapping identifiziert. Anschließend können die Metadaten anhand der folgenden Regeln überführt werden:

- Textuelle Inhalte werden 1:1 übernommen
- Metadaten mit einer Kardinalität größer 1 werden kommasepariert übernommen
- Referenzen auf Codelisten, werden falls möglich aufgelöst (z.B. URN auf Lizenz aus einer Liste von Lizenzen aus DCAT-AP.de)
- Geometrien werden falls möglich nur als WKT-Geometrien übernommen

Nach Überführung der Metadaten in das SDMX-CSV für referentielle Metadaten werden diese, analog zu automatischen Updates von Dataflows, auf die .Stat Suite hochgeladen. Referentielle Metadaten, die manuell während der Erstellung des Dataflows für einzelne Dimensionen erstellt wurde, werden dadurch nicht überschrieben. Dadurch können die allgemeinen Metadaten aus dem Metadatenkatalog separat von den konkreten Beschreibungen einzelner Datenpunkte behandelt werden.

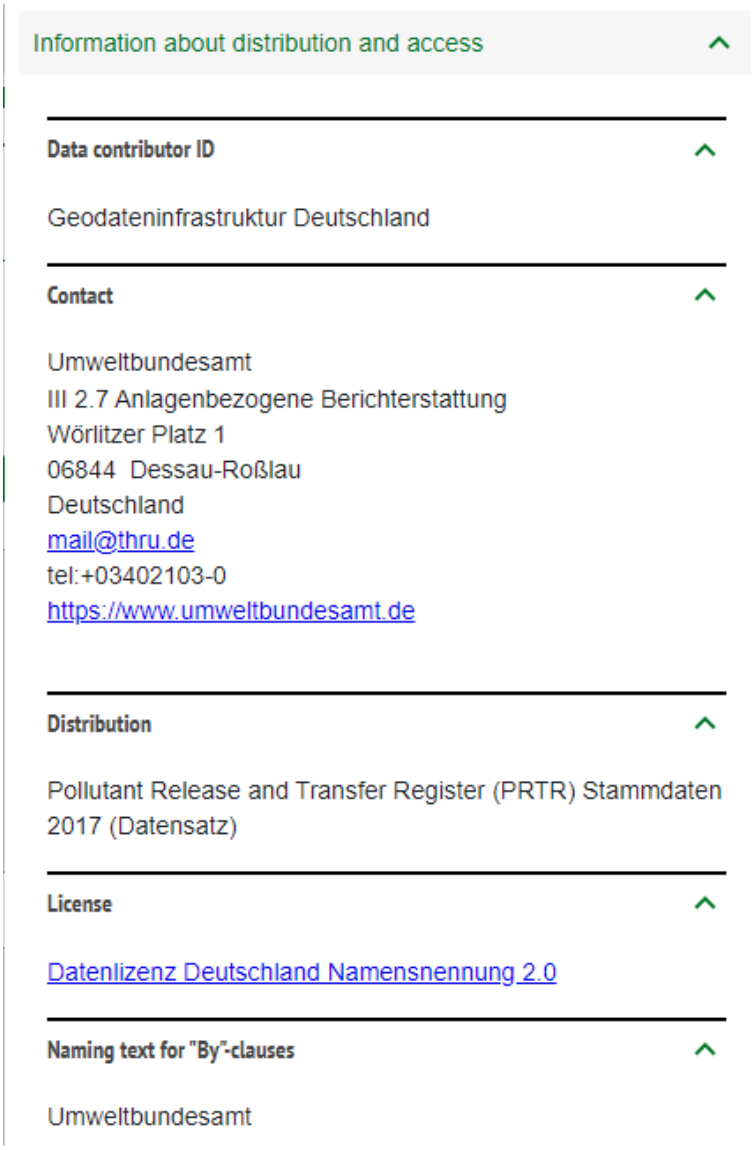
Abbildung 44 und Abbildung 45 zeigen die finale Darstellung der definierten Metadaten Attribute. In der ersten Abbildung sind alle Elemente zusammengeklappt, um einen Überblick über alle Kategorien zu erhalten. Kategorien ohne Daten werden automatisch ausgeblendet. Die zweite Abbildung zeigt Detailinformation einer Kategorie. Jedes einzelne Attribut kann bei Bedarf ebenfalls zusammengeklappt werden.

Abbildung 44: .Stat Suite Metadaten-Gruppierung im Side Panel, zusammengeklappt



Quelle: eigene Darstellung, con terra GmbH

Abbildung 45: .Stat Suite Metadaten-Gruppierung im Side Panel, aufgeklappt



Quelle: eigene Darstellung, con terra GmbH

5.3 Drupal Entwicklung

Die in der .Stat Suite verfügbaren Daten sollen auf der UBA Webseite nutzbar sein, wobei der Fokus auf der Erstellung von interaktiven Diagrammen liegt, die in Drupal-Artikel eingebunden werden können. Dabei sollen die Daten aus der .Stat Suite abgerufen werden damit keine manuelle Vorbereitung der Daten notwendig ist.

5.3.1 Auswahl der Technologien

Zunächst musste entschieden werden, welche Technologien zur Anzeige und Designen der Diagramme in Drupal integriert werden sollen. Als Anzeige-Technologie wurde Highcharts (Highsoft AS, o. J. - c) ausgewählt, da es bereits im UBA im Einsatz ist, sehr umfangreiche Funktionen zur Darstellung von Diagrammen bietet und das UBA diese Software bereits lizensiert hat.

Für das Design der Diagramme standen zwei Technologien zur Auswahl. Hier wurden nur Technologien betrachtet, die eine Online-Gestaltung von Highcharts-Diagrammen gestaltet, da ansonsten eine Einbettung des Editors in die Drupal-Webseite nicht möglich ist.

Die erste Option war der Highcharts-Editor (Highsoft AS, o. J. - d). Dieser ist speziell für Highcharts entworfen und kann so die Funktionalitäten von Highcharts optimal ausnutzen. Der Highcharts-Editor ist von Drupal unabhängig, weshalb für eine Nutzung im UBA-Drupal eine Integration konzipiert und umgesetzt werden muss.

Problematisch ist, dass der Highcharts-Editor zum Zeitpunkt der Entscheidungsfindung noch nicht final fertiggestellt war und auch die Arbeiten nicht fortgeführt wurden. Dies bedeutet, dass für einen Einsatz in UBA-Drupal einige Funktionen selbst korrigiert oder finalisiert werden müssten. Zudem war eine Verfügbarkeit von Updates und Sicherheits-Patches sehr unwahrscheinlich.

Die zweite Option war das Drupal-Plugin easychart (jyve, et al., o. J.). Dieses Plugin erweitert Drupal um die Möglichkeit Diagramme zu entwerfen und diese in Artikel einzubetten. Zur Darstellung der Diagramme können unterschiedliche Technologien eingesetzt werden, darunter auch Highcharts. Dieses Plugin wird bereits im UBA-Drupal eingesetzt, jedoch ohne Anbindung an die .Stat Suite, sodass auch hier eigene Entwicklungen konzipiert und umgesetzt werden müssten.

Da bei beiden Optionen Eigenentwicklungen notwendig wären wurde aufgrund folgender Vorteile das easychart Plugin ausgewählt:

- Verfügbarkeit von Updates und Patches
- Die easychart Nutzer müssen keine neue grafische Oberfläche erlernen
- Ältere (nicht .Stat Suite) Diagramme können weiterhin in Drupal genutzt werden
- Ältere (nicht .Stat Suite) Diagramme und neuere .Stat Suite Diagramme haben ein einheitliches Erscheinungsbild

5.3.2 Meeting Strukturen

Während des Umsetzungszeitraums wurden wöchentliche Videokonferenzen zwischen der Projektleitung des UBA und dem Entwicklungsteam der con terra (technische Leitung und Entwickler) abgehalten. Bei Bedarf wurden weitere Akteure eingeladen, wie z.B.:

- Projektleitung auf Seiten der con terra
- Technische Leitung der Datenintegration
- Ansprechpartner zum Betrieb von Drupal (werk21)
- Drupal-Anwender vom UBA

In diesen Telefonkonferenzen wurde der Entwicklungsfortschritt besprochen und neue Arbeitspakete abgestimmt und priorisiert.

In einem weiteren regelmäßigen Termin haben sich die Akteure der con terra aus den Teilbereichen Datenintegration und Drupal-Entwicklung ausgetauscht, um die Kompatibilität der beiden Bereiche zu gewährleisten und mögliche Synergien zu schöpfen.

5.3.3 Source Code Verwaltung und Ticket System

Die Source Code Verwaltung und das Ticket System für die Drupal-Integration sind bei opencode.de als eigenes Projekt angelegt (Umweltbundesamt, con terra GmbH, o. J.). Dieses

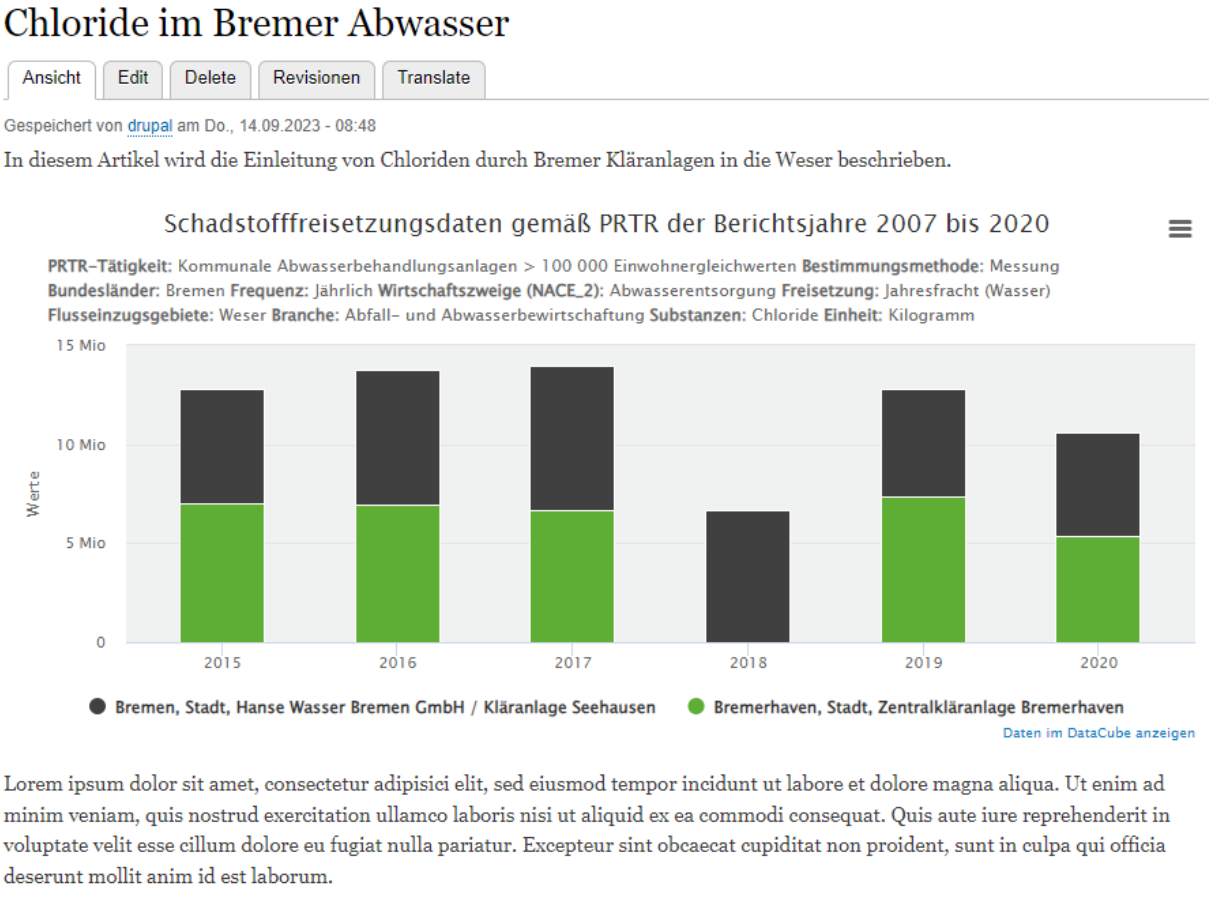
Projekt wird vom UBA verwaltet, die Übergabe des Quellcodes der Entwicklungen geschieht damit automatisch. Zur Zeit der Entwicklung war das Projekt nicht öffentlich verfügbar.

Die Anpassungen an dem easychart Plugin sind unter der Open Source Lizenz „GNU General Public License, Version 2“ (GPLv2) (Free Software Foundation (FSF), 1991 - b) veröffentlicht. Dies erfüllt die Lizenzanforderungen des originalen easychart Plugins und von Drupal, welche jeweils auch unter der GPLv2 lizensiert sind.

5.3.4 Anpassungen und Erweiterungen an easychart

Neben der Integration in Drupal, zusammen mit werk21, wurden einige Anpassungen und Erweiterungen an easychart vorgenommen. Diese sind im Folgenden beschrieben.

Abbildung 46: Beispiel der easychart Integration in Drupal



Quelle: eigene Darstellung, con terra GmbH

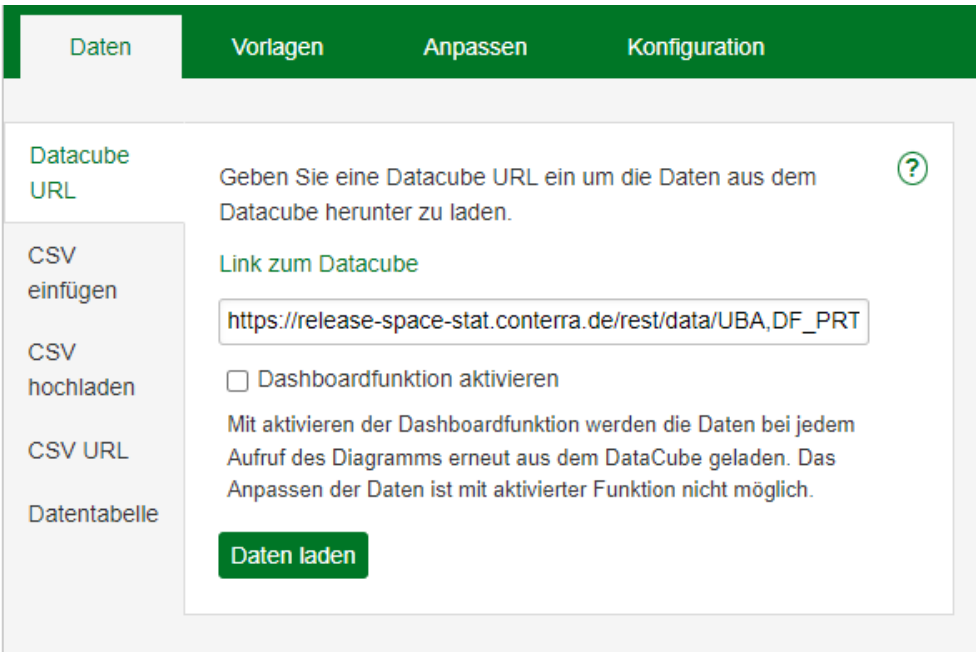
5.3.4.1 Anbindung an die .Stat Suite

Die wichtigste neue Funktion ist die Anbindung an die .Stat Suite. Über eine Daten-URL, die aus dem .Stat Suite Data Explorer kopiert wird, können die Daten einer Datenreihe in easychart eingelesen werden. Diese stehen dort zur Erzeugung eines Diagramms bereit. Eine detaillierte Anleitung zur Erzeugung dieser URL ist eingebettet.

Bei der Datenabfrage können zwei verschiedenen Modi genutzt werden. Standardmäßig werden die Daten nur einmalig abgefragt. Diagramme, die auf diesen Daten beruhen, ändern sich nicht, auch wenn es Änderungen an den zugrundeliegenden Daten in der .Stat Suite gibt. Dies ist

besonders für in Artikel eingebettete Diagramme sinnvoll, da der Artikel-Text ggf. auf Diagramminhalte verweist.

Abbildung 47: Eingabe der .Stat Suite URL im easychart Editor



Quelle: eigene Darstellung, con terra GmbH

Alternativ kann die Dashboardfunktion aktiviert werden. Ist dies der Fall, werden die Diagramm-Daten jedes Mal neu von der .Stat Suite angefragt, wenn die Webseite (z.B. der Drupal-Artikel) neu geladen wird. Dadurch können in Drupal Dashboards mit automatisch aktualisierten Daten erstellt werden. Dies kann insbesondere mit Datenreihen, die einen dynamischen Zeitraum nutzen (z.B. Daten der letzten 6 Monate), sinnvoll sein.

Unterhalb der Diagramme wird automatisch ein Link eingefügt, mit dem der Nutzer die originalen Datenreihen in der .Stat Suite aufrufen kann.

Der Diagrammtitel und der Untertitel werden zusätzlich aus der .Stat Suite ausgelesen. Diese Werte können vom Nutzer in Drupal angepasst werden.

5.3.4.2 Unterstützung von mehreren Diagrammen in einem Drupal-Artikel

Aufgrund einer ungenügenden Trennung der Webseitendaten im easychart Modul, kam es bei dem originalen Plugin zu Fehlern, wenn mehrere Diagramme in einem Drupal-Artikel eingesetzt wurden. Dabei haben die Datenreihen aus einem Diagramm die Datenreihen der anderen überschrieben. Dieses Problem wurde behoben.

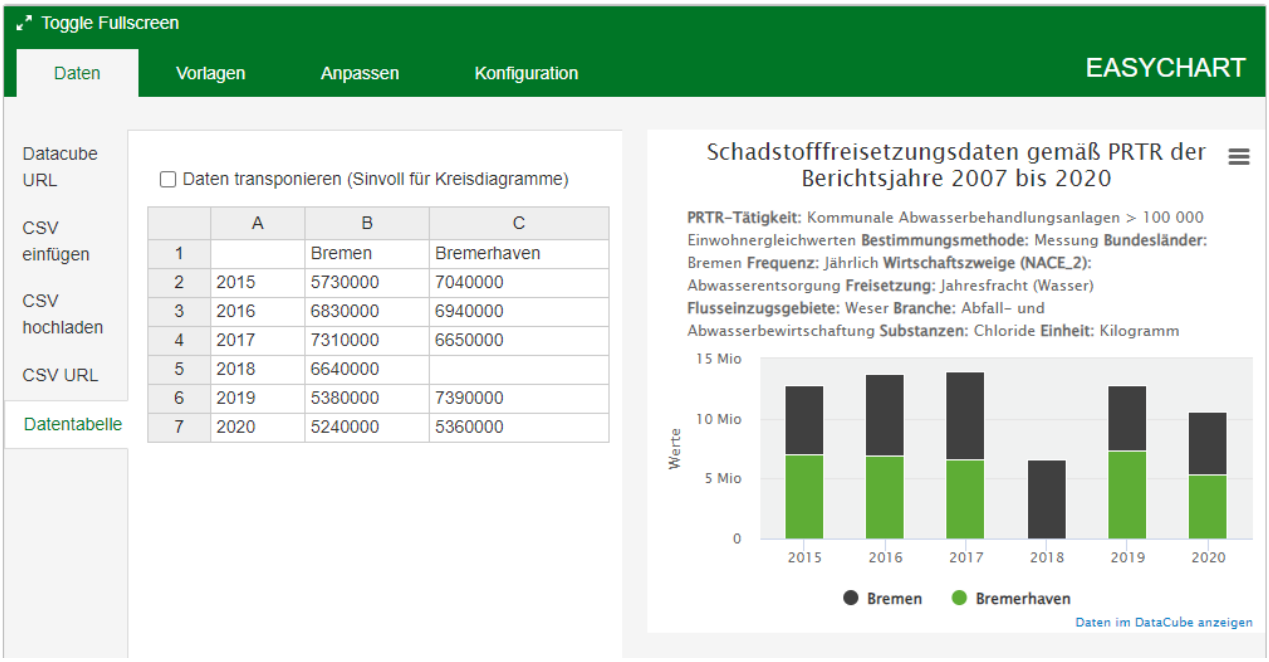
5.3.4.3 Unterstützung von Mehrsprachigkeit

Die Funktionen zur Mehrsprachigkeit von Drupal wurden auf das easychart Plugin angewendet. Dadurch können insbesondere auch die Metadaten (z.B. Titel der Datenreihe und Legenden) je nach Sprache der Anwendenden aus der .Stat Suite abgerufen und verwendet werden.

5.3.4.4 Transformation der Daten-Tabellen

Für einige Darstellungen ist es notwendig die Daten-Tabellen zu transformieren, also Spalten und Zeilen zu tauschen. Das easychart Plugin wurde um diese Funktion erweitert.

Abbildung 48: easychart Editor mit Datentabelle und Vorschau



Quelle: eigene Darstellung, con terra GmbH

5.3.4.5 Anpassung und Vereinheitlichung des Layouts

Das Layout und die Darstellung der Diagramme wurde an die Anforderungen des UBA angepasst. Dies umfasst die Integration von Hinweistexten, Textgrößen und -einzügen sowie die Auswahl der Hintergrundfarben.

5.3.5 Testbereitstellungen

Die Tests durch das UBA erfolgten in zwei Stufen. In der ersten Stufe wurden die Entwicklungen in den regelmäßigen Videokonferenzen präsentiert und konnten vom UBA-Projektleiter inspiziert werden. In der zweiten Stufe wurde das angepasste easychart Plugin von werk21 in die Drupal-Testumgebung integriert. Hier konnte das easychart Plugin von verschiedenen Nutzern des UBA geprüft werden.

6 Weitere Entwicklungsmöglichkeiten

Zu Beginn der Projektlaufzeit wurde eine Anforderungsanalyse mit anschließender Konzeption durchgeführt. Im zweiten Teil folgte die Umsetzung des Konzeptes, konkret, die Inbetriebnahme der .Stat Suite als Lösungskomponente mit der Übernahme verschiedener Datensätze des UBA. Im Folgenden werden verschiedene Themen andiskutiert, die nach der Projektlaufzeit bearbeitet werden könnten.

Während der Projektlaufzeit konnten Datensätze aus verschiedenen UBA Themengebieten erfolgreich in die .Stat Suite importiert werden. Damit der Data Cube auch in Zukunft erfolgreich in Wert gesetzt werden kann, müssen die bestehenden Datensätze regelmäßig aktualisiert werden. Kapitel 5.2.2.3.2 beschreibt das Vorgehen für automatische Dataflow Updates. Im Projektverlauf ist aufgefallen, dass für sehr viele Datenquellen keine Schnittstelle bereitsteht, die eine automatische Aktualisierung ermöglicht. Hierfür gibt es verschiedene Gründe, einige Datenquellen sind über deren API zum Beispiel nur unvollständig abrufbar. Für andere Datensätze gibt es keine Schnittstelle, oder diese ist auf Infrastrukturellen Gründen nicht automatisch von einem öffentlich erreichbaren Server abrufbar (zum Beispiel interne Datenbanken). Dies bedeutet, dass einige Datensätze nur durch manuelle Arbeit aktualisiert werden können. Sollten sich neue Schnittstellen ergeben, könnten die jeweiligen Dataflows auch in Zukunft automatisiert werden. Zusätzlich können weiterhin neue Datensätze durch das etablierte Verfahren (Kapitel 5.2) für den Data Cube aufbereitet und integriert werden.

In Kapitel 5.2.2.2 wurde die automatisierte Erstellung von SDMX-Strukturen erläutert. Diese basiert derzeit auf der Nutzung von Konfigurationsdateien. Durch die Verwendung dieser Dateien wurde das benötigte Wissen über die SDMX-XML bereits erheblich reduziert und Codelists und Dataflows können anhand der Konfigurationen durch FME automatisch generiert werden. Um den Prozess weiterzuentwickeln wäre es denkbar, eine Webanwendung zu implementieren, die Nutzende interaktiv bei der Konfiguration unterstützt. Für die Dataflow-Matrix wäre es zum Beispiel möglich, Konzepte und Codelists durchsuchbar zu machen und diese direkt aus Vorschlaglisten auszuwählen. Im Zuge dessen könnten auch Validierungen wie z.B. die einfache Vergabe einer ID integriert werden. Weitere Möglichkeiten wären die Erstellung von Codelists/Konzepten sowie die Konfiguration automatisierter Dataflow-Updates. Eine solche Entwicklung sollte auf einer eigenen UX-Analyse basieren und konnte im Projektverlauf aus zeitlichen Gründen nicht realisiert werden.

Zuletzt kann die Synchronisation von Metadaten zwischen UBA-Metadatenkatalog und Data Cube (siehe Kapitel 5.2.2.4.3) erst finalisiert werden, wenn die Umsetzungsphase des Metadatenkataloges abgeschlossen ist und dieser mit der entsprechenden API zur Verfügung steht. Bis dahin können Metadaten jedoch manuell über die CSV-Dateien der referentiellen Metadaten der .Stat Suite gepflegt werden.

Während der Entwicklung und den Tests der Diagramm-Integration in Drupal (siehe Kapitel 5.3) hat sich gezeigt, dass für einige Anwendungsfälle auch eine tabellarische Darstellung der Daten aus der .Stat Suite interessant ist. Dies ist über die eingesetzte Highcharts Bibliothek nicht möglich, sodass eine eigene Implementierung der Tabellenansicht notwendig ist. Dies konnte allerdings nicht mehr zur Projektlaufzeit umgesetzt werden.

7 Fazit

Das Data Cube Projekt verfolgt das Ziel, den Aufwand zur Bereitstellung der "Daten zur Umwelt" zu reduzieren und gleichzeitig die Möglichkeiten der Auswertung zum Beispiel durch interaktive Explorationen zu flexibilisieren. Durch die Verwendung neuer Technologien sollte zudem die Voraussetzung für weitergehende Anwendungen und Auswertungen, zum Beispiel durch KI, geschaffen werden. Zuletzt sollte Verzahnung mit umwelt.info bedacht und umgesetzt werden. Arbeitsgegenstände hierzu waren die Konzeption und Implementierung der Komponenten Data-Store, Data-Input, Data-Explorer und Data-Output.

Zur Umsetzung der Ziele wurde zunächst eine Anforderungsanalyse durchgeführt, durch die der redaktionelle Prozess im UBA erarbeitet wurde. Anschließend wurden verschiedene Lösungskomponenten analysiert. Letztlich wurde die .Stat Suite gewählt und ein Lösungsansatz erarbeitet, welcher im Anschluss umgesetzt und in Betrieb genommen werden konnte.

Durch die verschiedenen Komponenten der .Stat Suite konnten die identifizierten Anforderungen umgesetzt werden. Die SDMX-basierte Datenhaltung ermöglicht die Integration und Homogenisierung von Daten verschiedenster Strukturen. Da es sich bei SDMX um einen internationalen Standard handelt, ist das verwendete Datenmodell bereits ausführlich dokumentiert und verschiedene Anleitungen und Beschreibungen stehen online zur Verfügung. Dazu eignet sich der .Stat DLM sehr gut zur Verwaltung der Daten und Datenstrukturen. Die .Stat Suite kann Daten und deren Strukturen versioniert vorhalten und regelt über das Rechte-/Rollenkonzept und die Data Spaces die Zugriffe, so dass die Verwaltung und das Betrachten der Daten nur durch autorisierte Nutzende möglich ist. Durch die .Stat Data-Explorer Komponente können Datensätze interaktiv durchsucht und betrachtet werden. Dabei können Nutzende eigenständig Datensätze filtern, sortieren und sowohl als Tabelle, als auch durch verschiedene Diagrammtypen graphisch darstellen. Zur Anbindung von weiteren Systemen ist der Zugriff auf die Daten über eine API möglich. Hiermit ist eine einfache Möglichkeit geschaffen, die Daten technologieneutral auch für andere Anwendungsfälle zu verwenden. Weiterhin können die Tabellen und Diagramme aufwandsarm als iFrames in andere Anwendungen eingebettet werden. Zuletzt wurde FME als Lösungskomponente zur Integration von Daten und Automatisierung verschiedener Arbeitsschritte etabliert. Sowohl das SDMX-Format als auch die Quellformate verschiedener Daten konnten somit erfolgreich verarbeitet werden. Darüber hinaus konnte erfolgreich ein Drupal-Plugin implementiert werden, um die Daten aus der .Stat Suite an verschiedenen Stellen innerhalb von Drupal zu integrieren. Dies ermöglicht eine nahtlose Integration in Berichte, ohne auf iFrames oder Verlinkungen zurückgreifen zu müssen. Die Standard-Visualisierungen der .Stat Suite wurden durch diese Erweiterung deutlich aufgewertet.

Als open-source Lösung mit einer aktiven Community, mit Beteiligung von großen Organisationen wie OECD oder unicef, kann von einer fortlaufenden Weiterentwicklung der Software ausgegangen werden. Weiterhin ist die .Stat Suite bereits für den Betrieb via Docker oder Kubernetes ausgelegt, wodurch die Infrastruktur zukunftsorientiert aufgebaut werden kann.

Optimierungspotential besteht bei der automatisierten Anbindung von Datensätzen. Grund hierfür ist, dass viele Datenquellen nicht über eine strukturierte API verfügen, oder die Datensätze nur in internen Datenbanken zur Verfügung stehen, auf die aus Infrastrukturellen Gründen kein Zugriff besteht.

Innerhalb des Data-Explorers der .Stat Suite fehlt die Möglichkeit Daten zur Laufzeit neu zu berechnen (z.B. Einheitenkonvertierungen, Aggregationen). Diese Anforderungen konnte jedoch durch die entsprechende Vorprozessierungen zumindest teilweise erfüllt werden.

Der fachliche Aufwand zur Abstimmung einheitlicher Data Cube Strukturen hat sich als sehr groß herausgestellt. Ein simpler Import der Daten der unterschiedlichsten Daten ist zwar ohne weiteres möglich, der große Mehrwert eines Data Cube entsteht jedoch durch einheitliche Codelists um vergleichbare Datensätze zu identifizieren und die Exploration der Daten dadurch zu vereinfachen. Dieser Aufwand kann voraussichtlich reduziert werden, sobald für alle Themenfelder des UBA initial Daten in das System eingespielt wurden und somit eine Basis an einheitlichen Strukturen existiert.

Zur Verknüpfung von umwelt.info und Data Cube wurde ein Konzept entworfen, wie Metadaten in Zukunft aus dem Metadatenkatalog zum Data Cube synchronisiert werden, um beide Projekte in Wert zu setzen. Hierzu wurde die technische Transformation von DCAT-AP.de zu SDMX-CSV bereits umgesetzt. Die Produktivsetzung steht aus, bis der Metadatenkatalog fertig umgesetzt ist.

Insgesamt wird durch die Umsetzung des Data Cube Projektes die Bereitstellung von Daten im Kontext "Daten zur Umwelt" deutlich verbessert Daten liegen nun strukturiert in einer Datenbank vor und können interaktiv durchsucht und in der Data-Explorer Komponente dargestellt werden. Die Bereitstellung von manuell erzeugten Excel-Dateien über die UBA-Webseite kann dadurch entfallen. Während Daten zur Anzeige in Diagrammen zuvor manuell in den Easycharts Editor kopiert werden mussten, können Diagramme nun direkt aus der Datenhaltung der .Stat Suite generiert werden. Bei Bedarf werden Diagramme automatisch mit neuen Daten aktualisiert, wodurch weiterer manueller Aufwand entfällt.

8 Quellenverzeichnis

Australian Early Development Census - AEDC. (o. J.). *Data Explorer*. Retrieved 09 29, 2021, from https://www.aedc.gov.au/data/data-explorer

Avbar, R., Norton, P., & Chadwick, M. (o. J.). *Tableau Dashboard Integration*. Retrieved 11 16, 2021, from Drupal.org: https://www.drupal.org/project/tableau_dashboard

Bank for International Settlements - BIS. (o.J.). Retrieved 11 29, 2023, from https://www.bis.org/innovation/bis_open_tech_sdmx.htm

Bundesanstalt für Gewässerkunde. (o. J.). *Wasser-DE*. Retrieved 10 05, 2021, from https://www.wasser-de.de/

Bundesministerium des Innern, für Bau und Heimat. (2017). *Leitlinie für Informationssicherheit in der Bundesverwaltung.* Retrieved 11 22, 2021, from https://www.bmi.bund.de/SharedDocs/kurzmeldungen/DE/2017/09/up-bund.html

Carbon Disclosure Project - CDP. (o. J.). *CDP Cities, States and Regions Open Data Portal*. Retrieved 09 09, 2021, from https://data.cdp.net

con terra GmbH. (o. J.). *FME Server*. Retrieved 02 17, 2022, from https://www.conterra.de/portfolio/fme/fme-server

Der Beauftragte der Bundesregierung für Informationstechnik. (2020). *Architekturrichtlinie für die IT des Bundes.* Retrieved 10 08, 2021, from https://www.cio.bund.de/SharedDocs/Publikationen/DE/Architekturen-und-Standards/architekturrichtlinie_it_bund_techn_spezif_2020.pdf;jsessionid=850B89811404F5661A91CDB8A0FC0AA4.2_cid340?__blob=publicationFile

Deutsche Rohstoffagentur, Bundesanstalt für Geowissenschaften und Rohstoffe. (o. J.). *ROSYS - Rohstoffinformationssystem*. Retrieved 09 29, 2021, from https://rosys.dera.bgr.de/mapapps49prev/resources/apps/rosys2/index.html?lang=de

Drupal. (o. J.). *tableau_dashboard*. Retrieved 11 24, 2021, from Drupal Git Repository: https://git.drupalcode.org/project/tableau_dashboard

Europäische Kommission. (o. J. - a). *EDGAR - Emission Database for Global Atmospheric Research*. Retrieved 09 29, 2021, from https://edgar.jrc.ec.europa.eu

Europäische Kommission. (o. J. - b). *Euro SDMX Registry*. Retrieved 02 08, 2022, from https://webgate.ec.europa.eu/sdmxregistry/

Europäische Kommission. (o. J. - c). *Europäische Datenstrategie*. Retrieved 09 29, 2021, from https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/european-data-strategy_de

Europäische Kommission. (o. J. - d). *Introduction to EU Login*. Retrieved 02 17, 2022, from https://webgate.ec.europa.eu/cas/about.html

Europäische Kommission. (o. J. - e). *SDMX Converter*. Retrieved 02 08, 2022, from https://webgate.ec.europa.eu/sdmxconverter/convert

Europäische Kommission. (o. J. - f). *SDMX Converter - In a nutshell*. Retrieved 02 08, 2022, from https://ec.europa.eu/eurostat/cros/content/sdmx-converter-0_en

Europäische Kommission. (o. J. - g). *SMDX Converter - User Guide*. Retrieved 02 08, 2022, from ec.europa.eu: https://ec.europa.eu/eurostat/cros/content/sdmx-converter-user-guide_en

Federal Geographic Data Committee. (o. J.). *ISO 191** Suite of Geospatial Metadata Standards*. Retrieved 02 01, 2024, from FGDC.gov: https://www.fgdc.gov/metadata/iso-suite-of-geospatial-metadata-standards

Food and Agriculture Organization of the United Nations - FAO. (o. J.). *FAOSTAT*. Retrieved 10 19, 2021, from https://www.fao.org/faostat/en/#data

Free Software Foundation (FSF). (1991 - a). *GNU General Public License, Version 2*. Retrieved 11 24, 2021, from https://www.gnu.org/licenses/old-licenses/gpl-2.0.html

Free Software Foundation (FSF). (1991 - b). *GNU General Public License, Version 2*. Retrieved 12 5, 2023, from https://www.gnu.org/licenses/old-licenses/gpl-2.0.html

Geschäfts- und Koordinierungsstelle GovData - GKSt. (2022). *DCAT-AP.de Spezifikation 2.0*. Retrieved 12 11, 2023, from https://www.dcat-ap.de/def/dcatde/2.0/spec/#abstract

GitLab Inc. (o. J.). *Feature Branch Workflow*. Retrieved 11 27, 2023, from https://docs.gitlab.com/ee/gitlab-basics/feature_branch_workflow.html

Heinz, J. (2020). *TechStack go! Neue Plattform Generation einsatzbereit*. (Seven2one Informationssysteme GmbH) Retrieved 11 25, 2021, from seven2one.de: https://www.seven2one.de/news/techstack-go-neue-plattform-generation-einsatzbereit/

Highsoft AS. (o. J. - a). *Chart types*. Retrieved 11 22, 2021, from Highcharts.com: https://www.highcharts.com/docs/chart-and-series-types/chart-types

Highsoft AS. (o. J. - b). *Drill down*. Retrieved 11 22, 2021, from Highcharts.com: https://www.highcharts.com/docs/chart-concepts/drilldown

Highsoft AS. (o. J. - c). *Highcharts Webseite*. Retrieved 12 5, 2023, from https://www.highcharts.com/

Highsoft AS. (o. J. - d). *Highcharts® Editor*. Retrieved 11 22, 2021, from https://www.highcharts.com/products/highcharts-editor/

Highsoft AS. (o. J. - e). *Zooming*. (Highsoft) Retrieved 11 22, 2021, from Highcharts.com: https://www.highcharts.com/docs/chart-concepts/zooming

Highsoft AS. (o. J. - f). *Labels and string formatting*. (Highsoft) Retrieved 11 22, 2021, from Highcharts.com: https://www.highcharts.com/docs/chart-concepts/labels-and-string-formatting

IBM. (o. J.). *Stakeholder Map*. Retrieved 09 29, 2021, from https://www.ibm.com/design/thinking/page/toolkit/activity/stakeholder-map

jyve, De Jaeger, K., Daem, T., Kannekens, P., Vanderstukken, B., & Baele, F. (o. J.). *easychart*. Retrieved 12 5, 2023, from Drupal.org: https://www.drupal.org/project/easychart

Keycloak Authors, The Linux Foundation. (o. J.). *KEYCLOAK - Open Source Identity and Access Management*. Retrieved 11 22, 2021, from https://www.keycloak.org/

Komm.ONE. (o. J.). *Open CoDe*. Retrieved 11 27, 2023, from https://opencode.de/de

Luber, S. (2017). *Was ist ein OLAP Cube?* (Vogel IT-Medien GmbH) Retrieved 10 08, 2021, from BIGDATA-INSIDER: https://www.bigdata-insider.de/was-ist-ein-olap-cube-a-654603/

Mansmann, S., Ur Rehman, N., Weiler, A., & Scholl, M. (2014). Discovering OLAP dimensions in semi-structured data. *Information Systems*, pp. 120-133.

Martin, K., & Osterlin, M. (2013). *Value Stream Mapping: How to Visualize Work and Align Leadership for Organizational Transformation.* McGraw-Hill Education Ltd.

Microsoft. (o. J.). *Visual Studio Code*. Retrieved 11 30, 2023, from https://code.visualstudio.com/

Miro. (o. J.). *miro*. Retrieved 09 29, 2021, from https://www.miro.com

Object Management Group. (2005). *What is UML?* Retrieved 10 08, 2021, from https://www.uml.org/what-is-uml.htm

Organisation for Economic Co-operation and Development - OECD. (o. J.). *OECD Data*. Retrieved 09 29, 2021, from https://data.oecd.org

Redaktion ComputerWeekly.de, TechTarget, Inc. (o. J.). *Online Analytical Processing (OLAP)*. Retrieved 10 08, 2021, from ComputerWeekly.com: https://www.computerweekly.com/de/definition/Online-Analytical-Processing-OLAP

Rudolf, H. (2018). *Umweltdatenmanagement – Eine Geo-Inspiration.* Karlsruhe: Bernhard Harzer Verlag GmbH.

Rudolf, H. (2020). envVisio mit neuen Ansätzen im Umweltdatenmanagement: modelltheoretisch hergeleitet, fachlich ausgearbeitet, praktisch umgesetzt. In *Umweltinformationssysteme – Wie verändert die Digitalisierung unsere Gesellschaft?* Wiesbaden: Springer Fachmedien.

Safe Software Inc. (2020). *FME Article - Getting Started with Gallery Apps*. (FME Community) Retrieved 02 17, 2022, from https://community.safe.com/s/article/getting-started-with-gallery-apps

Safe Software Inc. (2022). *FME Article - Getting Started with FME Server Workspace Apps*. (FME Community) Retrieved 02 17, 2022, from https://community.safe.com/s/article/fme-server-apps

Safe Software Inc. (o. J. - a). *FME - CSV (Comma Separated Value) Writer Parameters*. Retrieved 02 17, 2022, from https://docs.safe.com/fme/html/FME_Desktop_Documentation/FME_ReadersWriters/csv2/csv2_writer.htm

Safe Software Inc. (o. J. - b). *FME - Matrix View of FME Integrations Gallery*. Retrieved 02 17, 2022, from https://www.safe.com/fme/formats-matrix/

Safe Software Inc. (o. J. - c). *FME - What is Data Transformation?* Retrieved 02 17, 2022, from https://www.safe.com/what-is/data-transformation/

Safe Software Inc. (o. J. - d). *FME - What is Data Validation?* Retrieved 02 17, 2022, from https://www.safe.com/what-is/data-validation/

Safe Software Inc. (o. J. - e). *FME Server - Data Integration and Automation*. Retrieved 02 17, 2022, from https://www.safe.com/fme/fme-server/

SDMX Technical Standards Working Group - SDMX TWG. (2021). *SDMX-CSV*. Retrieved 02 09, 2022, from SDMX TWG GitHub Website: https://github.com/sdmx-twg/sdmx-csv/blob/master/data-message/docs/sdmx-csv-field-guide.md

Seven2one. (o. J.). Leitfaden für den Anwender - seven2one Mesap. Handbuch im Kontext des ZSE.

Sisense. (o. J. - a). *Adding drill hierarchies to widgets*. Retrieved 11 23, 2021, from Sisense.com: https://documentation.sisense.com/docs/adding-drill-hierarchies-to-widgets

Sisense. (o. J. - b). *Adding Widgets to a Dashboard*. Retrieved 11 18, 2021, from Sisense.com: https://documentation.sisense.com/docs/adding-widgets-to-a-dashboard

Sisense. (o. J. - c). *Drilling down in a Widget*. Retrieved 11 18, 2021, from Sisense.com: https://documentation.sisense.com/docs/drilling-down-in-a-widget

Sisense. (o. J. - d). *ElastiCubes*. Retrieved 11 18, 2021, from Sisense.com: https://documentation.sisense.com/docs/elasticubes

Sisense. (o. J. - e). *Embedding*. Retrieved 11 18, 2021, from Sisense.dev: https://sisense.dev/guides/embedding/

Sisense. (o. J. - f). *Finding Tables and Columns*. Retrieved 11 18, 2021, from Sisense.com: https://documentation.sisense.com/docs/finding-tables-and-columns

Sisense. (o. J. - g). *Interacting with Filters as a Viewer*. Retrieved 11 18, 2021, from Sisense.com: https://documentation.sisense.com/docs/interacting-with-filters-as-a-viewer-dashboard

Sisense. (o. J. - h). *Introduction to Data Sources*. Retrieved 11 18, 2021, from Sisense.com: https://documentation.sisense.com/docs/introduction-to-data-sources

Sisense. (o. J. - i). *Share Dashboards*. Retrieved 11 18, 2021, from Sisense.com: https://dtdocs.sisense.com/article/share-dashboards

Sisense. (o. J. - j). *Widgets*. Retrieved 11 18, 2021, from Sisense.dev: https://sisense.dev/guides/js/plugins/widgets.html

Statistical Data and Metadata eXchange - SDMX Community. (2011 - c). *SDMX Information Model*. Retrieved 02 14, 2022, from https://sdmx.org/wp-content/uploads/SDMX_2-1-1_SECTION_2_InformationModel_201108.pdf

Statistical Data and Metadata eXchange - SDMX Community. (2021 - a). *SDMX 3.0 Technical Specifications*. Retrieved 02 14, 2022, from https://sdmx.org/?page_id=5008

Statistical Data and Metadata eXchange - SDMX Community. (2021 - b). *SDMX 3.0 Technical Specifications*. Retrieved 11 18, 2021, from https://sdmx.org/?page_id=5008

Statistical Data and Metadata eXchange - SDMX Community. (2021 - d). *SDMX Standards - Summary of major changes and new functionality*. Retrieved 02 14, 2022, from https://sdmx.org/wp-content/uploads/SDMX_3-0-0_Major_Changes_FINAL-1_0.pdf

Statistical Data and Metadata eXchange - SDMX Community. (2021 - e). *SDMX Standards: Section 1.* Retrieved 11 29, 2021, from https://sdmx.org/wp-content/uploads/SDMX_3-0-0_SECTION_1_FINAL-1_0.pdf

Statistical Data and Metadata eXchange - SDMX Community. (o. J. - f). *The official site for the SDMX community*. Retrieved 10 07, 2021, from https://sdmx.org/

Statistical Information System Collaboration Community - SIS-CC. (2020). *.Stat Academy Webinar - .Stat Suite installation using docker compose*. (SIS-CC) Retrieved 01 30, 2024, from https://www.youtube.com/watch?v=U2knnqOr5ws

Statistical Information System Collaboration Community - SIS-CC. (2021 - an). *How to install Stat Suite using docker compose in less than 10 minutes*. (SIS-CC) Retrieved 01 30, 2024, from YouTube.com: https://www.youtube.com/watch?v=9D4Q9K33JJg

Statistical Information System Collaboration Community - SIS-CC. (o. J. - a). *.Stat Data Explorer*. (SIS-CC) Retrieved 02 07, 2022, from https://siscc.org/stat-suite/stat-data-explorer/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - aa). *.Stat Suite documentation - Toolbar*. (SIS-CC) Retrieved 02 15, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/toolbar/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ab). *.Stat Suite documentation - Upload data*. (SIS-CC) Retrieved 11 18, 2021, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/upload-data/).

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ac). *.Stat Suite documentation - Viewing data*. (SIS-CC) Retrieved 02 08, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ad). *.Stat Suite documentation Git Repository*. (SIS-CC) Retrieved 02 07, 2022, from SIS-CC GitLab: https://gitlab.com/sis-cc/dotstatsuite-documentation/-/tree/master/static/images

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ae). *.Stat Suite FAQ*. Retrieved 11 24, 2021, from https://siscc.org/faq/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - af). *Become a member*. (SIS-CC) Retrieved 02 16, 2022, from https://siscc.org/contact/contact-become-a-member/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ag). *Copy data structures*. Retrieved 02 09, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/copy-data-structures/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ah). *Data Cube - Betrieb Repository (GitLab)*. Retrieved 11 20, 2023, from https://gitlab.opencode.de/uba-data-cube/data-cube-betrieb/-/tree/cc389ebd40271fc69543c75b58287aa5907c2e8a

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ai). *Delete data (GitLab)*. Retrieved 02 09, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/delete-data/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - aj). *DotStat Transfer Service (GitLab)*. Retrieved 11 30, 2023, from https://gitlab.com/sis-cc/.stat-suite/dotstatsuite-core-transfer

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ak). *dotstatsuite-data-viewer (GitLab)*. Retrieved 02 09, 2022, from https://gitlab.com/sis-cc/.stat-suite/dotstatsuite-data-viewer

Statistical Information System Collaboration Community - SIS-CC. (o. J. - al). *General layout and common features - URL Parameters*. Retrieved 02 15, 2022, from SIS-CC GitLab: https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/general-layout/#url-parameters

Statistical Information System Collaboration Community - SIS-CC. (o. J. - am). *Governance & funding*. (SIS-CC) Retrieved 02 16, 2022, from https://siscc.org/who-we-are/governance/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ao). *Information Side Panel*. Retrieved 01 05, 2024, from .Stat Suite documentation: https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/preview-table/information-panel/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ap). *Infrastructure requirements*. Retrieved 02 09, 2022, from .Stat Suite documentation: https://sis-cc.gitlab.io/dotstatsuite-documentation/getting-started/infrastructure-requirements/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - aq). *Manage user access*. Retrieved 02 09, 2022, from .Stat Suite documentation: https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/manage-user-access/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - ar). *Members*. (SIS-CC) Retrieved 02 16, 2022, from siscc.org: https://siscc.org/who-we-are/members/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - as). *Partnerships*. (SIS-CC) Retrieved 02 16, 2022, from siscc.org: https://siscc.org/who-we-are/partners/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - at). *Upload data*. Retrieved 02 09, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/upload-data/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - au). *Upload data structures*. Retrieved 02 09, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/upload-structure/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - av). *Youtube Kanal der SIS-CC*. (SIS-CC) Retrieved 02 16, 2022, from https://www.youtube.com/channel/UCZGlYlrmeb1MbLONpxObGUQ

Statistical Information System Collaboration Community - SIS-CC. (o. J. - aw). *tachyon Release Milestone*. Retrieved 02 01, 2024, from .Stat Suite Git Repository: https://gitlab.com/groups/sis-cc/.stat-suite/-/milestones/64#tab-issues

Statistical Information System Collaboration Community - SIS-CC. (o. J. - b). *.Stat Suite Docker-Compose - git repository*. (SIS-CC) Retrieved 11 22, 2021, from https://gitlab.com/sis-cc/.stat-suite/dotstatsuite-docker-compose

Statistical Information System Collaboration Community - SIS-CC. (o. J. - c). *.Stat Suite documentation*. (SIS-CC) Retrieved 11 18, 2021, from https://sis-cc.gitlab.io/dotstatsuite-documentation/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - d). *.Stat Suite documentation*. Retrieved 11 29, 2023, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/preview-table/custom-data-view/default-selection/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - e). *.Stat Suite documentation - .Stat Core module*. (SIS-CC) Retrieved 11 23, 2021, from https://sis-cc.gitlab.io/dotstatsuite-documentation/getting-started/framework/#stat-core-module

Statistical Information System Collaboration Community - SIS-CC. (o. J. - f). *.Stat Suite documentation - .Stat DE configuration*. (SIS-CC) Retrieved 02 08, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/configurations/de-configuration/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - g). *.Stat Suite documentation - .Stat DE customisation*. (SIS-CC) Retrieved 11 19, 2021, from https://sis-cc.gitlab.io/dotstatsuite-documentation/configurations/de-customisation/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - h). *.Stat Suite documentation - .Stat RESTful Web Service Cheat Sheet*. (SIS-CC) Retrieved 02 15, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-api/restful/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - i). *.Stat Suite documentation - .Stat Suite Open Source Framework*. (SIS-CC) Retrieved 11 22, 2021, from https://sis-cc.gitlab.io/dotstatsuite-documentation/getting-started/framework/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - j). *.Stat Suite documentation - Charts*. (SIS-CC) Retrieved 02 15, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/charts/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - k). *.Stat Suite documentation - Common header & footer for all views*. (SIS-CC) Retrieved 02 15, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/common-header-and-footer/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - l). *.Stat Suite documentation - Customise chart layout*. (SIS-CC) Retrieved 02 15, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/charts/customise-feature/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - m). *.Stat Suite documentation - Customise table layout*. (SIS-CC) Retrieved 02 15, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/preview-table/customise-feature/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - n). *.Stat Suite documentation - Design principles & functional vision*. (SIS-CC) Retrieved 11 19, 2021, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/design-principles/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - o). *.Stat Suite documentation - Facets*. (SIC-CC) Retrieved 11 19, 2021, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/searching-data/facets/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - p). *.Stat Suite documentation - Free text search*. (SIS-CC) Retrieved 02 08, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/searching-data/free-text-search/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - q). *.Stat Suite documentation - Frequency & Time-Period*. (SIS-CC) Retrieved 02 14, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/filters/time-period/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - r). *.Stat Suite documentation - General layout and common features*. Retrieved 11 19, 2021, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/general-layout/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - s). *.Stat Suite documentation - Getting started*. (SIS-CC) Retrieved 02 16, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/getting-started/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - t). *.Stat Suite documentation - Implicit and explicit orders*. (SIS-CC) Retrieved 02 08, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/custom-data-view/implicit-explicit-order/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - u). *.Stat Suite documentation - Indexing data*. (SIS-CC) Retrieved 02 07, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/searching-data/indexing-data/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - v). *.Stat Suite documentation - Installation using Docker-Compose*. (SIS-CC) Retrieved 11 23, 2021, from https://sis-cc.gitlab.io/dotstatsuite-documentation/install-docker/docker-compose/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - w). *.Stat Suite documentation - List related structures*. (SIS-CC) Retrieved 11 22, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-dlm/list-related-data-structures/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - x). *.Stat Suite documentation - Notes displaying attributes in table views*. (SIS-CC) Retrieved 02 23, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/preview-table/footnotes/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - y). *.Stat Suite documentation - Search results*. (SIS-CC) Retrieved 02 08, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/searching-data/search-results/

Statistical Information System Collaboration Community - SIS-CC. (o. J. - z). *.Stat Suite documentation - Share*. (SIS-CC) Retrieved 02 15, 2022, from https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/viewing-data/share/

Tableau Software, LLC. (o J. - d). *Faster analytics with Hyper*. Retrieved 11 29, 2021, from https://www.tableau.com/products/new-features/hyper

Tableau Software, LLC. (o. J. - a). *Cube-Datenquellen*. (TABLEAU SOFTWARE) Retrieved 11 15, 2021, from https://help.tableau.com/current/pro/desktop/de-de/cubes.htm

Tableau Software, LLC. (o. J. - b). *Das Tableau Datenmodell*. (TABLEAU SOFTWARE) Retrieved 11 23, 2021, from https://help.tableau.com/current/pro/desktop/de-de/datasource_datamodel.htm

Tableau Software, LLC. (o. J. - c). *Einbetten von Ansichten in Webseiten*. (TABLEAU SOFTWARE) Retrieved 11 23, 2021, from Tableau.com: https://help.tableau.com/current/pro/desktop/de-de/embed.htm

Tableau Software, LLC. (o. J. - e). *Freigeben von Webinhalten*. (TABLEAU SOFTWARE) Retrieved 11 23, 2021, from Tableau.com: https://help.tableau.com/current/pro/desktop/de-de/shareworkbooks.htm

Tableau Software, LLC. (o. J. - f). *Gastbenutzer*. (TABLEAU SOFTWARE) Retrieved 11 16, 2021, from Tableau.com: https://help.tableau.com/current/server/de-de/users_guest.htm

Tableau Software, LLC. (o. J. - g). *Preise für datenorientierte Anwender*. (TABLEAU SOFTWARE) Retrieved 11 24, 2021, from Tableau.com: https://www.tableau.com/de-de/pricing/embedded

Tableau Software, LLC. (o. J. - h). *Tableau - Grundlegendes zu Lizenzmodellen und Product Keys*. (TABLEAU SOFTWARE) Retrieved 11 24, 2021, from Tableau.com: https://help.tableau.com/current/server/de-de/license_product_keys.htm

Tableau Software, LLC. (o. J. - i). *Tableau - Preisinformationen*. (TABLEAU SOFTWARE) Retrieved 11 24, 2021, from Tableau.com: https://www.tableau.com/de-de/pricing/teams-orgs

Tableau Software, LLC. (o. J. - j). *Tableau Desktop*. (TABLEAU SOFTWARE) Retrieved 11 16, 2021, from Tableau.com: https://www.tableau.com/de-de/products/desktop

Tableau Software, LLC. (o. J. - k). *Tableau Prep*. (TABLEAU SOFTWARE) Retrieved 11 15, 2021, from Tableau.com: https://www.tableau.com/de-de/products/prep

Tableau Software, LLC. (o. J. - l). *Tableau Prep - Datenquellen*. (TABLEAU SOFTWARE) Retrieved 11 15, 2021, from Tableau.com: https://www.tableau.com/de-de/products/prep#data-sources

Tableau Software, LLC. (o. J. - m). *Tableau Server*. (TABLEAU SOFTWARE) Retrieved 11 16, 2021, from Tableau.com: https://www.tableau.com/de-de/products/server

Umweltbundesamt. (2020 - a). *Umsetzungskonzept für ein Umwelt- und Naturschutzinformationssystem (UNIS-D) – ein nutzer- und anwendungsorientiertes Angebot der Umweltverwaltungen – Leistungsbeschreibung.* Dessau-Roßlau.

Umweltbundesamt. (2020 - b). *Umwelt beobachten*. Retrieved 09 29, 2021, from https://www.umweltbundesamt.de/das-uba/was-wir-tun/forschen/umwelt-beobachten#was-ist-umweltbeobachtung

Umweltbundesamt. (2020 - c). *Umwelt- und Naturschutzinformationssystem UNIS-D – Machbarkeitsstudie.* Dessau-Roßlau.

Umweltbundesamt. (o. J.). *Daten zur Umwelt*. Retrieved 09 29, 2021, from https://www.umweltbundesamt.de/daten

Umweltbundesamt, con terra GmbH. (o. J.). *easychart-drupal-modul*. Retrieved 12 05, 2023, from opencode.de: https://gitlab.opencode.de/uba-data-cube/easychart-drupal-modul

United Nations - UN. (o. J.). *UN Population Division Data Portal*. Retrieved 09 29, 2021, from https://population.un.org/dataportal/home

Wikipedia. (o. J.). *Data Lake*. Retrieved 10 07, 2021, from https://de.wikipedia.org/wiki/Data_Lake

A Anhang

A.1 Excel-Liste der Anforderungen an die Komponenten des Data Cube

./Anhang/Anforderungen_DataCube.xlsx

A.2 Beschreibung ausgewählter IST-Datensätze der "Daten zur Umwelt"

./Anhang/Analyse_datenhaltende_Stellen.zip

A.3 Excel-Liste der reduzierten Anforderungen für die Tool-Vorauswahl

./Anhang/ToolVorauswahl/Tool_Vorauswahl_Matrix.xlsx

A.4 Zusammenfassung von Stärken und Schwächen von vorausgewählten Tools

./Anhang/ToolVorauswahl/Tool_Vorauswahl.docx