Eberhard Karls Universität Tübingen

Mathematisch-Naturwissenschaftliche Fakultät

Fachbereich Informatik

Studiengang Medieninformatik

Masterarbeit

Open Data Collaboration

Thomas Pentenrieder, Matrikelnummer 3554566

Betreuer: **Kevin Koerner**

Prüfer: **Prof. Dr. Thomas Walter**

Prof. Dr. Torsten Grust

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angabe von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Masterarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Thomas Pentenrieder (Matrikelnummer 3554566), 31.10.2016

Abstract

In dieser Arbeit wird das Thema *Open Data* beleuchtet und dabei speziell auf die derzeit verfügbaren Werkzeuge zur Verwaltung und Publikation offener Daten eingegangen. Im Kern steht dabei die Beobachtung, dass keines der untersuchten Open Data-Tools dazu geeignet ist, dass auch technisch nicht-versierte Menschen gemeinschaftlich an strukturierten Daten arbeiten, diese sammeln, bearbeiten und unter Open Data-Richtlinien weitergeben können. Als möglicher Lösungsansatz hierfür wird nachfolgend ein Konzept für eine Webapplikation vorgeschlagen, über die es mithilfe eines visuellen Formulareditors möglich ist, Datenstrukturen zu definieren, ohne dass hierfür Datenbankkenntnisse nötig sind. Basierend auf einer so konstruierten Struktur können über ein simples Webformular Daten hinzugefügt und bearbeitet werden. Für veröffentlichte Datensätze können zudem von anderen Personen Änderungsvorschläge (z.B. Korrekturen falscher Informationen, Hinzufügen fehlender Einträge) gemacht werden, die vom Besitzer moderiert werden. Um eine weitere maschinelle Verarbeitung der Daten zu erleichtern, ist zudem eine JSON-basierte Programmierschnittstelle verfügbar. Die Ermöglichung des gemeinschaftlichen Kuratierens von Daten im Sinne von *Open Data* ist das Hauptziel dieser Arbeit.

Danksagung

Ein großer Dank geht an meine Betreuer Kevin und Robert, die mir stets mit konstruktiver Kritik und guten Ratschlägen beiseite standen.

Ich möchte mich auch bei meiner Freundin Lena für die moralische Unterstützung und bei meinen Eltern bedanken, die mir dieses Studium ermöglicht haben.

Ein weiterer Dank geht außerdem alle Korrekturleser für die wertvollen Hinweise sowie an all diejenigen, die sich die Zeit genommen haben, den Prototyp zu testen und den Fragebogen auszufüllen.

Inhaltsverzeichnis

1. Einleitung 7

2. Was sind Daten 10

2.1 Arten von Daten 11

2.2 Datenformate 14

2.3 Linked Data 18

2.4 Small Data & Big Data 22

3. Open Data 27

3.1 „Open“-Bewegungen 27

3.2 Definition: Open Data 29

3.3 Lizenzen 35

3.4 Vorteile von Open Data 36

3.5 Bedenken gegen Open Data 39

3.6 Open Data Tools 41

4. Projekt Konzeption 46

4.1 Ziel des Projektes 46

4.2 Beispiel für ein Open Data-Projekt 46

4.3 Herausforderungen & Lösungsansätze 47

4.4 Verwendete Technologien 49

5. Projekt Umsetzung 55

5.1 Eingesetzte Technologieprodukte 55

5.2 Prototyp 59

6. Diskussion 64

7. Ausblick 68

8. Zusammenfassung 70

Abbildungsverzeichnis

Abbildung 1: Knowledge Pyramid - adaptiert von [2] 11
Abbildung 2: Beispiel für CSV Notation 16
Abbildung 3: Beispiel für XML-Notation 17
Abbildung 4: Beispiel für JSON Notation 18
Abbildung 5: Beispiel für Linked Data im RDF+XML Format 19
Abbildung 6: Visualisierung des RDF+XML Markups als Graph 20
Abbildung 7: Beispiel für Linked Data mit RDFa Markup 21
Abbildung 8: Beispiel für Linked Data in JSON-LD Notation 21
Abbildung 9: SPARQL Abfrage 22
Abbildung 10: State of the LOD Cloud (2014), lod-cloud.net [21] 22
Abbildung 11: 5-Sterne-Modell für Offene Daten [39] 34
Abbildung 12: Screenshot von CKAN Bearbeitung eines Datensatzes 43
Abbildung 13 Infobox auf Wikipedia, Daten aus Wikidata [79] 45
Abbildung 14: Gefahrentafel mit Gefahrnummer (oben) und UN-Nummer (unten) [78] 46
Abbildung 15: JSON Dokument entsprechend Abbildung 16 52
Abbildung 16: JSON Schema Beschreibung für JSON Dokument (Abbildung 15) 53
Abbildung 17: Architektur des Prototyps (Technologien) 54
Abbildung 18: Formular entsprechend dem JSON Schema aus Abbildung 16 58
Abbildung 19: Architektur des Prototyps (Produkte) 59
Abbildung 20: Erstellen eines Datensatzes mit Form Builder 60
Abbildung 21: Auflistung aller im Datensatz enthaltenen Einträge 61
Abbildung 22: Moderationsansicht für ausstehende Änderungen 62
Abbildung 23: Formular zum Hinzufügen und Bearbeiten von Daten 62
Abbildung 24: Dokumentation der Schnittstelle (API) als swagger Markup 63

1. Einleitung

> *„Digitales Gold, Rohstoff des 21. Jahrhunderts, neues Öl – mit diesen Begriffen wird das wirtschaftliche Potential von Daten gegenwärtig verglichen. Sollten die Datenbestände im Besitz des Staates nicht ebenso wertvoll sein und als offene Verwaltungsdaten zur Verfügung stehen, um einen volkswirtschaftlichen Mehrwert zu erzeugen?“ [1]*

Mit diesen Worten leiten die Autoren ihre 2016 im Auftrag der Konrad-Adenauer-Stiftung e.V. veröffentlichte Studie *Open Data. The Benefits - Das volkswirtschaftliche Potential für Deutschland* ein. Diese Studie versucht den aus den USA kommenden Trend der *Offenen Daten* (Open Data), d.h. Daten, die von jedermann ohne Einschränkungen verwendet werden können, auf Deutschland bezogen zu quantifizieren. Die Forscher kommen hierbei zu dem Schluss, dass durch die aktuelle Strategie der Veröffentlichung von Regierungsdaten in Deutschland ein wirtschaftliches Potenzial von 12,1 Mrd. Euro pro Jahr innerhalb der nächsten zehn Jahre erreicht werden kann. Würden die Anstrengungen diesbezüglich verstärkt, sehen die Autoren sogar ein Potenzial von 43,1 Mrd. bis hin zu 131,1 Mrd. Euro pro Jahr z.B. durch Firmenneugründungen mit auf den Daten basierenden, neuartigen Geschäftsmodellen oder Investoren, die durch die frei zugänglichen Informationen bessere Risikoanalysen durchführen können [1].

Die Breitstellung von Open Data ist allerdings nicht nur für Regierungen und Verwaltungen ein spannendes Thema, sondern auch für Nichtregierungsorganisationen und Privatpersonen sowie Firmen von zunehmendem Interesse. In diesem Zusammenhang beschäftig sich diese Arbeit sowohl mit den theoretischen Grundlagen von Open Data als auch mit den derzeit bestehenden Werkzeugen im Open Data-Ökosystem. Zuletzt wird ein neues Konzept für eine Datenplattform vorgestellt, die das gemeinschaftliche Sammeln, Verwalten und Veröffentlichen von Daten möglichst einfach und ohne spezielle Fachkenntnisse gestalten soll.

Als Hauptquelle für den theoretischen Teil wird das Buch *The Data Revolution* von Rob Kitchin herangezogen. Der Professor und *European Research Council Advanced Investigator* an der *National University of Ireland Maynooth* ist Autor zahlreicher Fachbücher und wissenschaftlicher Artikel zu den Themen *Data*, *Code* und *Human Geography*. Zusätzlich werden zahlreiche zum Thema passende wissenschaftliche Arbeiten und Bücher sowie Nachrichten und Blogeinträge an geeigneter Stelle referenziert.

Aufbau der Arbeit

Als Grundlage wird zunächst erläutert, was man unter dem Begriff *Daten* versteht, dessen etymologische Herkunft und wie dieser z.B. im Bereich der Informatik verwendet wird. Hierauf aufbauend wird das Konzept von Datenformaten betrachtet und speziell im Bereich von *Open Data* relevante Vertreter dieser vorgestellt. Zudem gibt es einen Überblick über das Thema *Linked Data*, welches ebenfalls eine wichtige Grundlage für *Open Data* bildet. Zum Abschluss dieses Kapitels folgt ein kurzer Exkurs zu *Big Data* und *Small Data*, um Einsatzbereich und Nützlichkeit des weiter unten vorgestellten Projekts besser einordnen zu können.

Hierauf aufbauend wird das Thema Open Data behandelt. Hierzu werden zunächst zur besseren Einordnung verwandte Bewegungen betrachtet, die den *Open*-Gedanken in sich tragen, wie z.B. *Open Source* oder *Open Government*. Der Fokus dieses Kapitels liegt auf den existierenden Definitionen von Open Data, wobei hier vor allem die Auslegungen der *Open Knowledge Foundation* (*Open Definition*), von opengovdata.org (*8 Principles of Open Government Data*) und von Tim Berners-Lee (*5 Star Open Data*) aufgegriffen und als Grundlage für den weiteren Text verwendet werden. Es folgt ein kurzer Abschnitt bezüglich der lizenzrechtlichen Aspekte sowie eine Ausführung zu Argumenten von Befürwortern und Skeptikern bzw. Kritikern von Open Data.

Zuletzt werden in diesem Kapitel bestehende Open Data-Werkzeuge vorgestellt und ihre Funktionen, Anwendungsgebiete sowie Nützlichkeit und Simplizität in der Benutzung diskutiert. Dies liefert die Grundlage für das im folgenden vorgeschlagene Konzept eines neuen Tools, welches sich speziell dem Thema der Kollaboration bei der Erstellung und Pflege von Open Data-Projekten widmet. Dieses wird in Kapitel 4 zunächst anhand der angestrebten Ziele sowie eines konkreten Anwendungsfalls näher erläutert.

Für die hierauf folgende Auflistung der technischen Herausforderungen inklusive der jeweils vorgeschlagenen Lösungsansätze werden die eingesetzten Technologien sowie die konkret verwendeten Produkte sowie die Gesamtarchitektur des Prototyps vorgestellt. Die programmiertechnisch umgesetzte Version des neuen Tools bzw. dessen Funktionsweise wird anhand von Screenshots erläutert. Anhand von gesammeltem Feedback folgt eine Auswertung bezüglich der zuvor gesetzten Ziele sowie eine Auflistung von hieraus abgeleiteten, möglichen Änderungen und Verbesserungen.

Abschließend wird betrachtet, wie das neu geschaffene Werkzeug möglicherweise helfen kann, den Open Data-Gedanken weiterzutragen und das Thema einer größeren Zielgruppe zugänglich zu machen. Eine kurze Zusammenfassung bezüglich der gegebenen Problemstellung und der erreichten Ziele rundet diese Arbeit ab.

2. Was sind Daten

Das Wort *Daten* ist der Plural von Datum, abstammend vom lateinischen *datum*, zu Deutsch *gegeben* (dare = geben, ausfertigen, schreiben) [1]. Daten werden einem in diesem Sinne also gegeben, als messbares oder speicherbares Erzeugnis von Geschehnissen. Dies steht allerdings im Widerspruch zur von Rob Kitchin beschriebenen, generellen Auffassung, dass Daten *genommen* werden, „*extrahiert durch Beobachtungen, Berechnungen, Experimente und Aufzeichnungen*“ ([2] S.2). Entsprechend müssten Daten eigentlich *Capten* (abgeleitet vom lateinischen *capere*, zu Deutsch *nehmen*) heißen.

Abgesehen von der etymologischen Herkunft des Wortes zeigt diese Einordnung bereits sehr gut, worum es sich bei Daten handelt: Sie sind stets eine Auswahl aus allen möglichen verfügbaren, quantifizierbaren Eigenschaften. Daten sind von Natur aus unvollständig, selektiv (sie zeigen nur einen gewählten Ausschnitt des Gesamtbildes) und repräsentativ (sie beschreiben einen Sachverhalt) ([2] S.3).

Vor allem bedingt durch die Wortherkunft beziehungsweise ihrer sehr allgemeinen Bedeutung ist die Interpretation und Verwendung des Wortes *Daten* sehr vielschichtig.

Daten sollten nicht mit Begriffen wie Fakten, Beweise, Informationen und Wissen gleichgesetzt werden, wenn es nach D. Rosenberg geht [3]. Diese bauen zwar auf Daten auf, sind allerdings untereinander zu unterscheiden.

> *„facts are ontological, evidence is epistemological, data is rhetorical. A datum may also be a fact, just as a fact may be evidence. But, from its first vernacular formulation, the existence of a datum has been independent of any consideration of corresponding ontological truth. When a fact is proven false, it ceases to be a fact. False data is data nonetheless.“* [3]

Daten sind aus dieser Sicht also immer Daten, unabhängig davon ob diese korrekt oder falsch und egal ob diese Teil von Fakten oder Beweisen sind. Sie bestehen, bevor sie durch Analyse und Interpretation zu solchen gemacht werden.

Das Verständnis von Daten ist allerdings auch stark abhängig von der Perspektive des Betrachters. Aus Sicht der Informatik sind Daten in erster Linie Nullen und Einsen, die elektronisch verarbeitet, übertragen und gespeichert werden können. Entsprechend dieser Definition sind nicht nur (strukturierter) Text, sondern auch Fotos, Videos und Musik als Daten anzusehen. Erst wenn diese aus ihrer binären Form berechnet werden, entstehen Informationen. Diese Perspektive vermischt allerdings das Konzept von Daten und die Art,

wie diese kodiert sind. Die Kodierung kann sowohl diskret (digital) als auch kontinuierlich (analog) sein. So können eine CD und eine Schallplatte die selbe Musik beinhalten, die Art der Speicherung ist jedoch grundlegend verschieden [4].

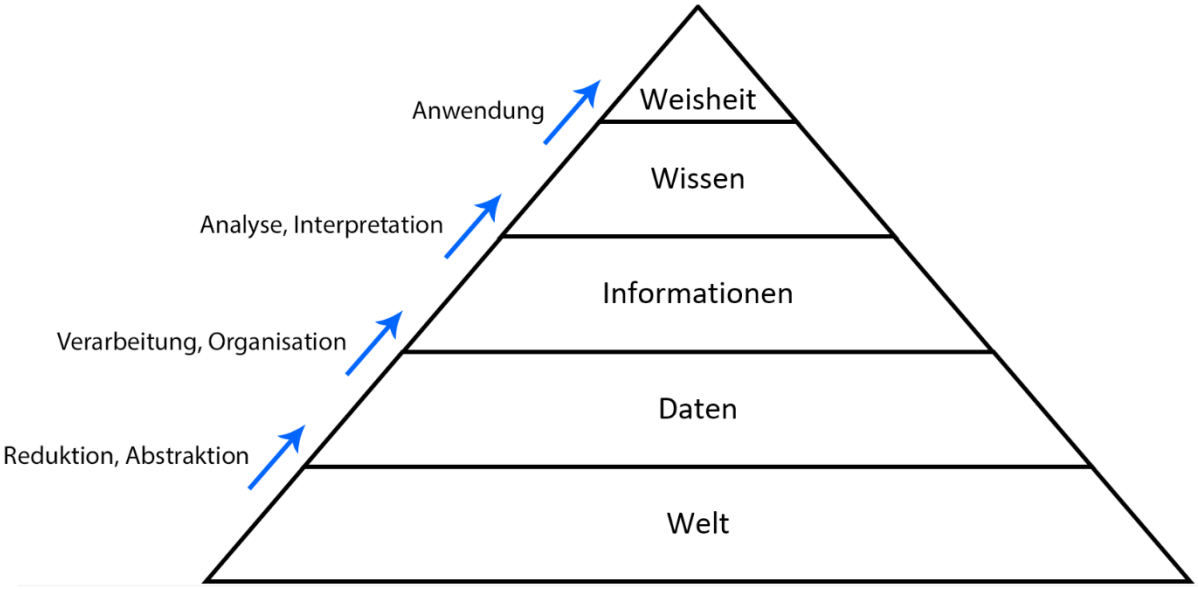

Abbildung 1: Knowledge Pyramid - adaptiert von [2]

Im Kern lassen sich Daten in den meisten Definitionen als Abstraktion unserer Umgebung ansehen, reduziert auf eine Dimension in der sie verarbeitet werden kann. Diese Daten können organisiert und verarbeitet werden, um aus den so entstandenen Informationen Schlussfolgerungen ziehen zu können.

2.1 Arten von Daten

Daten können anhand der Kriterien **Form** (qualitativ oder quantitativ), **Struktur** (strukturiert, teilweise strukturiert, unstrukturiert), **Quelle** (gesammelt, abgeleitet, aufgefangen, transient), **Produzent** (primär, sekundär, tertiär) und **Typ** (indexiert, attribuiert, Metadaten) charakterisiert werden, wie im Folgenden näher gezeigt wird ([2] S.4-9):

Form

Daten kommen in den verschiedensten Formen vor, z.B. als Zahlen, Symbole, Bilder, Text, Ton oder auch als elektromagnetische Wellen. Sie lassen sich in zwei Hauptkategorien unterscheiden:

Quantitative Daten bestehen aus numerischen Aufzeichnungen. Dies sind beispielsweise Messwerte (wie Höhen, Distanzen, Zeit, Mengen) oder Repräsentationen nicht-physischer

Zusammenhänge (Schulnoten, Ranglisten). Sie lassen sich visualisieren (Graphen, Diagramme), auswerten (in Form von Statistiken) und als Input für weitere Berechnungen verwenden.

Qualitative Daten sind hingegen nicht-numerisch, wie Texte, Bilder, Kunst, Ton und Videos. Diese können zwar in quantitative Daten umgewandelt werden, nicht jedoch ohne einen erheblichen Verlust des Umfangs der Quelldaten durch die Abstraktion und Reduktion auf numerische Werte.

Struktur

Daten unterscheiden sich zudem in ihrer Struktur bzw. bezüglich der Existenz oder Absenz einer solchen. Sie lassen sich in **strukturierte**, **teilweise strukturierte** und **unstrukturierte Daten** einordnen.

Strukturierte Daten folgen einem festen Schema (z.B. Klassen-Definitionen in der objektorientierten Programmierung) und können einfach gespeichert, übertragen und verarbeitet werden. Sie bestehen in der Regel aus Nummern und Text in konsistenten Formen (z.B. Namen, Adressen, Geburtsdaten, Telefonnummern) und können ohne größeren Aufwand als Input für Algorithmen verwendet sowie visuell aufbereitet und durchsucht werden.

Teilweise strukturierte Daten haben kein festes Schema und können so beispielsweise nicht in einer relationalen Datenbank gespeichert werden. Ihre Struktur ist unregelmäßig und oft verschachtelt, einzelne Teile sind aber selbstbeschreibend (z.B. durch entsprechend kenntlich gemachte Meta-Informationen) und kann so mit zusätzlichem Aufwand maschinell verstanden und verarbeitet werden. Beispielsweise folgen HTML-Dokumente keiner festen Struktur, die einzelnen HTML-Tags lassen allerdings auf die Art des Inhalts Rückschlüsse ziehen und können sowohl von Mensch als auch Maschine gelesen und verstanden werden (`<head>`, `<a>`, `<img/>`, `<article>`).

Unstrukturierte Daten folgen keinem allgemeingültigen Datenmodell. Zudem werden hier oft zahlreiche Datentypen und Formate vermischt (Text, Fotos, Videos etc.). Diese Art der Daten ist deutlich schwerer zu verarbeiten. Während bislang große Datensets vor allem aus strukturierten Daten bestanden, sorgen neue Technologien wie NoSQL zu Speicherung und Machine Learning zur Verarbeitung für ein enormes Wachstum von teilweise strukturierten

und unstrukturierten Daten. Schätzungsweise 80% aller Daten weltweit sind unstrukturiert, bei einem 15-fachen Wachstum gegenüber strukturierten Daten [5].

Generierung von Daten

Man unterscheidet grundlegend zwei unterschiedliche Arten der Generierung von Daten. Im Normalfall werden diese explizit **gesammelt**, durch Messungen, Beobachtungen, Formulare, Foto- und Videoaufnahmen etc. Ziel ist es, brauchbare Daten zu erhalten.

Im Gegensatz hierzu produzieren viele Systeme und Geräte zusätzlich zu ihrer Hauptfunktion weitere Daten als Abfallprodukt. Diese lassen sich ebenfalls **auffangen** und können durchaus wertvoll sein. Beispielsweise ist es die primäre Funktion von Fahrkartenautomaten, dem Passagier ein gültiges Ticket ausstellen zu können. Zudem lassen sich jedoch auch zahlreiche Rückschlüsse über die Frequentierung und die Hochlastzeitfenster eines Bahnhofes schließen, wenn man die entsprechenden Daten auffängt und auswertet.

Oft sind solche Nebenprodukte zu umfangreich, unstrukturiert oder uninteressant, so dass diese **transienten Daten** direkt wieder gelöscht werden, ohne dass sie analysiert, verarbeitet oder gespeichert werden. Dies kann beispielsweise bei Live-Videostreams der Fall sein, deren Speicherung zu aufwändig wäre und das Material im Nachhinein keinen Wert mehr hätte.

Die genannten Arten bezeichnet man als Rohdaten, die zunächst unverändert vorliegen. Im Gegensatz hierzu entstehen die **abgeleiteten Daten** durch weitere Verarbeitung, Konvertierung und Kombination mit anderen Daten. Beispielsweise könnte die oben erwähnte Sammlung von Daten eines Fahrkartenautomaten in Verbindung mit Datum und Uhrzeit statistische Auswertungen zur Nutzung des Bahnhofes in Ferienzeiten liefern.

Produzent

Je nach Produzent unterscheidet man zwischen primären, sekundären und tertiären Daten:

Primäre Daten sind die Daten aus erster Hand und werden z.B. von einem Forscher und seinen Gerätschaften erzeugt. **Sekundäre Daten** wurden vom Produzenten weitergegeben und können von anderen Personen genutzt, verarbeitet und analysiert werden. Dasselbe Material kann also gleichzeitig für den Erzeuger primäre und für jemand anderen sekundäre Daten sein.

Tertiäre Daten sind eine Form von abgeleiteten Daten, die keine direkten Rückschlüsse auf die Quelldaten zulassen, z.B. bei statistischen Auswertungen von Volksbefragungen (u.a. aus Anonymisierungsgründen).

Typen von Daten

Je nach Sinn und Bezug unterscheidet man zudem zwischen indexierten und attribuierten Daten und Metadaten:

Indexierte Daten enthalten eindeutige Kennzeichnungen (wie Passnummer, Nummernschilder, Kreditkartennummer, Seriennummern), anhand derer sie identifiziert und verlinkt werden können. Diese Art von Daten ist besonders einfach zu verarbeiten, da anhand der *Identifier* die Daten unterschieden, kombiniert, aggregiert, durchsucht und anderweitig verarbeitet werden können.

Attribuierte Daten enthalten Eigenschaften zu einer Sache, einem Ereignis oder ähnlichem, sind jedoch selbst nicht indexiert. Während beispielsweise ein Smartphone durch seine Seriennummer eindeutig identifiziert wird, kann diese Art von Daten weitere Auskunft über Größe, Gewicht, Betriebssystem, Displayart etc. geben.

Metadaten hingegen sind *Daten über Daten*, die sich entweder auf den Inhalt (z.B. Überschriften in Tabellen) oder ein gesamtes Datenpaket (z.B. Autor, Thema, Jahreszahlen etc.) beziehen können. Man unterscheidet hier *Beschreibende Metadaten*, die die Identifizierung und Auffindbarkeit von Datenpaketen erleichtern (Titel, Autor, Verleger, Beschreibung etc.), *Strukturelle Metadaten* zur Erläuterung des Umfangs und der internen Organisation der Daten und *Administrative Metadaten*, die beispielsweise Aufschluss über die Art der Datenerzeugung, verwendete Dateiformate oder lizenzrechtliche Einschränkungen geben.

2.2 Datenformate

Datenformate geben vor, wie digital vorliegende Daten von einer Software zu interpretieren sind [7]. Je nach Themenbereich und Anwendungsfall kann der Begriff allerdings unterschiedlich aufgefasst werden. In der Softwareentwicklung können Datenformate beispielsweise die Eigenschaften und Darstellung von Zahlen (Integer, Binary, Float etc.) definieren, während auf Anwendungsebene etwa das GIS-Datenformat eine standardisierte Beschreibung für die Speicherung von geographischen Daten ist. Datenformate können

beschreiben, wie die Binärdaten einer Datei interpretiert werden müssen, sie können sich allerdings auch auf die interne Struktur eines Textdokumentes (siehe CSV, XML, JSON) beziehen [8].

Weiter sind Dateiformate eine Untergruppierung der Datenformate, welche entsprechende Codierungsregeln für bestimmte Arten von Daten wie Text (z.B. RichText), Audio (MP3) oder Fotos (z.B. JPEG) festlegen. Wird ein solches Format von einer Person oder Firma definiert und die weitere Entwicklung kontrolliert, spricht man in der Regel von proprietären Dateiformaten. Ohne entsprechende Informationen zum Aufbau des Formats sind diese Daten für Außenstehende ohne Reverse Engineering, also dem meist aufwändigen Rekonstruieren der Struktur, nicht nutzbar. Zahlreiche proprietäre Dateiformate sind jedoch auch offen spezifiziert und können frei verwendet werden (z.B. Office Open XML).

Im Bereich von Open Data (siehe Abschnitt 3) werden naturgemäß offene Daten- und Dateiformate bevorzugt. Der Fokus liegt hierbei meist auf maschinell lesbaren, textbasierten Formaten, von denen drei wichtige Vertreter nachfolgend erläutert werden.

CSV (Comma-seperated values) beschreibt eine simple Methode zur Strukturierung von Daten im Textformat. Hierbei werden einzelne Datensätze und die hierin enthaltenen Felder durch ein festgelegtes Zeichen voneinander getrennt. Es gibt hierfür keinen offiziellen Standard, CSV ist aber als Memo bei der Internet Engineering Task Force (IETF) spezifiziert [6].

Datensätze werden in der Regel durch Zeilenumbrüche getrennt, sodass pro Zeile ein zusammengehöriger *Record* besteht. Die Felder einer Zeile werden oftmals durch ein Komma getrennt (daher der Name), abhängig von verwendeter Software, Präferenzen oder internen Standards finden allerdings auch andere Zeichen wie Semikolon, Doppelpunkt, Tabulator oder Leerzeichen als Trennzeichen Gebrauch. Zusätzlich kann die erste Zeile eines Dokumentes für Überschriften bei tabellenförmigen Daten verwendet werden.

Vorteil des CSV-Formates ist vor allem die kompakte Form, bei der sehr wenige zusätzliche Datenmengen (Overhead) durch das Markup entstehen, sondern lediglich ein Trennzeichen pro Feld und Zeile. Außerdem ist das Format durch die sehr simple, textbasierte Struktur unabhängig von Betriebssystemen oder Programmiersprachen nutzbar.

Problematisch bei CSV ist, dass viele Details zur korrekten Verarbeitung nicht mit im Dokument definiert sind (z.B. welches Feldbegrenzerzeichen verwendet wird, ob die erste Zeile aus Überschriften besteht oder diese auch schon zu den Daten gehört, welche Zeichencodierung verwendet wird). Die Schwierigkeit besteht darin, diese Informationen auf irgendeine Art und Weise anderen Beteiligten mitzuteilen. Zudem geht aus einem CSV-Dokument nicht hervor, in welcher Form beispielsweise Datum und Uhrzeit codiert sind oder ob bei Zahlen ein Punkt oder ein Komma als Dezimaltrennzeichen eingesetzt wird. Außerdem eignet sich das Format nur für Daten in Listen- beziehungsweise Tabellenform. Verschachtelte Strukturen müssen hierfür erst in eine flache Variante umgewandelt werden.

```
1  name,population,bundesland
2  Tübingen,89011,Baden-Württemberg
3  München,1388000,Bayern
```

Abbildung 2: Beispiel für CSV Notation

XML (Extensible Markup Language) ist eine Auszeichnungssprache für hierarchisch strukturierte, textbasierte Daten, die sowohl für Maschinen als auch Menschen lesbar ist. Diese wurde unter anderem vom World Wide Web Consortium (W3C) 1998 erstmals spezifiziert [7]. XML wird sowohl zur plattformunabhängigen Datenübertragung als auch zur Definition von Benutzeroberflächen und als Markup für Inhalte verwendet. Diverse Sprachen basieren auf XML, die durch spezielle Regeln für Struktur und Elemente unterschiedlichste Einsatzgebiete bedienen (z.B. RSS als News-Service, XAML für User Interfaces in .NET Sprachen, XHTML für Webseiten).

Typisch für die XML-Syntax sind die durch spitze Klammern gekennzeichneten Elemente, die entweder Text oder weitere Elementknoten beinhalten können. Im Gegensatz zu CSV kann XML selbstbeschreibend sein, da einzelne Knoten Metainformationen enthalten können. Zudem sollte zu Beginn jedes Dokuments die XML-Version sowie der verwendete Zeichensatz angegeben werden.

Der größte Nachteil von XML sind die zusätzlichen Datenmengen, die durch die Beschreibung der Daten hinzukommen. Vor allem durch die schließenden Tags (`</bundesland>`) wird das Dokument zusätzlich aufgebläht, ohne dass zusätzliche, brauchbare (Meta-)Informationen hierin enthalten wären. Allein im Beispiel von Abbildung 3: Beispiel für XML-Notation machen die schließenden Tags über 27% des gesamten Textes aus. Zudem wird mit

zunehmender Verschachtelung der Struktur, Vermischung von Tags und Attributen (`name="Tübingen"`), und der Verwendung von Kommentaren ein XML-Dokument für einen Menschen sehr schnell unübersichtlich.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <staedte>
3      <stadt name="Tübingen">
4          <population>89011</population>
5          <bundesland>Baden-Württemberg</bundesland>
6      </stadt>
7      <stadt name="München">
8          <population>1388000</population>
9          <bundesland>Bayern</bundesland>
10     </stadt>
11 </staedte>
```

Abbildung 3: Beispiel für XML-Notation

JSON (ursprünglich JavaScript Object Notation), ist ein textbasiertes, für Mensch und Maschine lesbares Datenformat, welches primär für Datenaustausch entwickelt wurde. Ursprünglich von Douglas Crockford als Subset von JavaScript konzipiert ist die Notation allerdings sprachunabhängig und es existieren entsprechende Parser für alle gängigen Programmiersprachen.

Aktuell gibt es zwei konkurrierende Spezifikationen angesehener Institutionen: RFC 7159 der Internet Engineering Task Force (IETF) [8] und ECMA-404 der Ecma Organisation [9]. Im Kern sind sich diese beiden allerdings einig bezüglich unterstützter Strukturen (z.B. Objekte, Arrays) und Datentypen (z.B. string, char, number) sowie der Formatierung (z.B Verwendung eckiger & geschweifter Klammern).

Dank der Unabhängigkeit von einzelnen Sprachen, der einfachen Lesbarkeit und des relativ geringen Overheads (vor allem im Vergleich zu XML) hat sich als eine der Standardnotationen für Datenübertragungen vor allem im Web (speziell bei REST-APIs) etabliert. Nachteil bei JSON ist, dass die Notation keinen Platz für zusätzliche Meta-Informationen für einzelne Elemente vorsieht. Hierfür müssen eigene Felder innerhalb der entsprechenden Objekte angelegt werden [10].

```
1  {
2      "staedte":
3      [{
4          "name": "Tübingen",
5          "population": 89011,
6          "bundesland": "Baden-Württemberg"
7      }, {
8          "name": "München",
9          "population": 1388000,
10         "bundesland": "Bayern"
11     }]
12 }
```

Abbildung 4: Beispiel für JSON Notation

2.3 Linked Data

Das Konzept von Linked Data beschreibt eine Methode zur Verknüpfung einzelner Daten untereinander. Der Begriff wurde 2006 vom „Erfinder des World Wide Web“ und Vorsitzender des World Wide Web Consortium (W3C) Tim Berners-Lee in einer Notiz zum „*Semantic Web*“ [11] geprägt (siehe auch Abschnitt 3.2). Hauptziel ist dabei, die Bedeutung von Daten und deren Beziehungen untereinander so aufzubereiten, dass diese für Maschinen verständlich sind.

Das Internet besteht heute zum Großteil aus Dokumenten, die für Menschen einfach zu verstehen sind. Wir können beispielsweise aus Fließtext relevante Informationen extrahieren, auf Bildern Personen und Objekte zuordnen oder aus Audio- und Videoaufnahmen Musikstücke erkennen. Dies ist trotz der Fortschritte in Bereichen wie maschinellem Lernen und Computer Vision immer noch eine enorm hohe technische Hürde – besonders, wenn diese Informationen in Echtzeit zum Beispiel in einer Suchmaschine zur Verfügung gestellt werden müssen.

Deshalb wird mit Linked Data versucht, möglichst kleine, eindeutige Informationsblöcke zu identifizieren und diese mittels Verweise auf andere Daten einzuordnen. Diese Blöcke werden durch *International Resource Identifiers* (IRI) eindeutig identifiziert. Dies ist in der Regel ein *Universal Resource Indicator* (URI), also ein Link zur Definition des Objekts.

Technisch gesehen werden solche Dateneinheiten in Form eines *Triples* beschrieben, also einer dreiteiligen Schreibweise:

- **Subjekt**

  Das zu beschreibende Subjekt kann z.B. eine Person, ein Gegenstand, ein Ort oder ein künstlerisches Werk sein, identifiziert durch eine IRI.
- **Prädikat**

  Das Prädikat (auch Property oder Relation genannt) beschreibt, wie sich das Subjekt zum Objekt verhält beziehungsweise welcher Sachverhalt diese beiden verbindet, z.B. Adam **kennt** Eva, Tübingen **liegt in** Baden-Württemberg oder Benjamin Blümchen **mag** Zuckerstückchen.
- **Objekt**

  Das Objekt kann entweder wie das Subjekt ein weiteres, beliebiges, eindeutig identifiziertes Datenobjekt sein oder ein Literal, also ein konkreter Wert wie zum Beispiel ein Name, ein Messwert oder ein historisches Datum.

Diese Triples werden in dem **Resource Description Framework (kurz RDF)** Format beschrieben. Hierbei handelt es sich um einen vom W3C verabschiedeten Standard [12] zur Modellierung von Daten für den Austausch über das World Wide Web. Daten, die mittels dem RDF-Modell beschrieben und verknüpft werden, bilden zusammen einen (gerichteten) Graphen, weshalb man manchmal auch von Graph Data anstelle von Linked Data spricht.

Dabei gibt es verschiedene Schreibweisen für RDF-Dokumente. Die älteste und auch am weitesten verbreitete Variante ist **RDF+XML**, die anfangs mangels Alternativen oft auch einfach nur RDF genannt wurde.

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <rdf:RDF
3          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4          xmlns:dbo="http://dbpedia.org/ontology#" >
5
6      <rdf:Description rdf:about="http://dbpedia.org/resource/Bavaria">
7          <dbo:population>12440000</dbo:population>
8          <dbo:capital rdf:resource="http://dbpedia.org/resource/Munich" />
9      </rdf:Description>
10
11 </rdf:RDF>
```

Abbildung 5: Beispiel für Linked Data im RDF+XML Format

Abbildung 5: Beispiel für Linked Data im RDF+XML Format zeigt ein simples Beispiel für Linked Data im RDF-XML Markup. Hier wird als erstes die XML-Version und die Codierung

des Dokumentes spezifiziert. Dies ist optional, wird von der RDF-XML-Spezifikation [13] allerdings empfohlen.

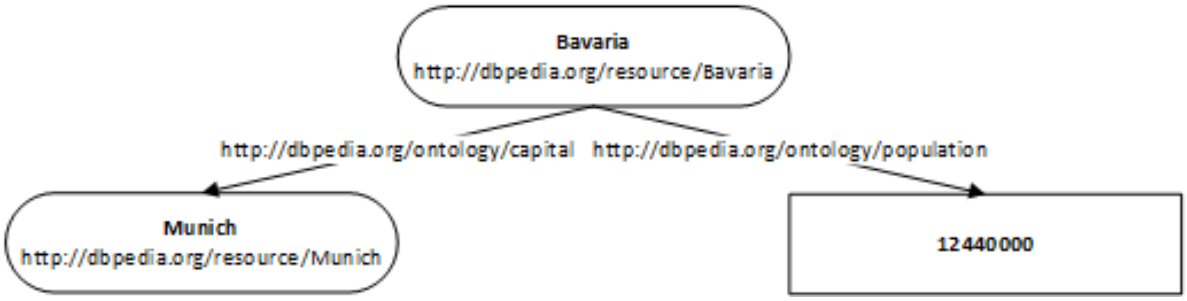

Abbildung 6: Visualisierung des RDF+XML Markups als Graph

Als Container wird anschließend ein **`<rdf:RFD>`** Objekt angelegt, in welchem an oberster Stelle die Namespaces (Zeile 3 & 4) definiert werden. Diese geben Auskunft darüber, an welcher Stelle die später verwendeten Prädikate definiert sind. In diesem Fall wurde der Standard-Namespace für RDF-Dokumente (beinhaltet Schlüsselwörter wie *description*, *resource* und *about*) sowie der Namespace von *dbpedia.org* importiert. Dieser Container kann nun beliebig viele Datenbeschreibungen enthalten – auch leer wäre dies bereits ein valides RDF-Dokument.

Mit **`<rdf:Description>`** beginnt die eigentliche Beschreibung einer Entität (das Subjekt), welche über die **`rdf:about`** Property eindeutig identifiziert ist – in diesem Fall das Bundesland Bayern entsprechend der *dbpedia.org* Seite. Eine Description kann nun mehrere Prädikat-Objekt-Kombinationen enthalten, mit denen sie jeweils ein Triple bildet.

Die hier verwendeten Prädikate stammen ebenfalls aus dem Namespace von *dbpedia.org*. **`<dbo:population>`** beinhaltet dementsprechend die Einwohnerzahl, die hier als einfache Zahl, also als Literal angegeben ist. Dagegen verknüpft **`<dbo:capital>`** das Bundesland Bayern mit seiner Hauptstadt München.

[14] [15]

Im Laufe der Zeit haben sich neben RDF+XML weitere Syntax-Varianten für das Resource Description Framework entwickelt und etabliert. Nachfolgend werden die gängigsten hiervon kurz anhand desselben Beispiels erläutert.

RDFa ist eine Erweiterung zu HTML5, mithilfe der in Webseiten enthaltene Daten vor allem für Suchmaschinen und sonstige Crawler hervorgehoben und verständlich gemacht werden können.

> *„RDFa is a way of publishing Linked Data in HTML5 documents — it solves both the data silo-ing and data linking problem"* [16]

Derzeit sind zwei RDFa-Varianten vom W3C standardisiert: die RDFa 1.1 Primer [17] und eine hiervon abgespeckte Version RDFa Lite 1.1 [18]. Abbildung 7: Beispiel für Linked Data mit RDFa Markup zeigt, wie das vorige Beispiel im RDFa Format aussehen könnte:

```
1  <p prefix="dbo: http://dbpedia.org/ontology#"
2     resource="http://dbpedia.org/resource/Bavaria">
3     <span property="dbo:capital" resource="http://dbpedia.org/resource/Munich">
4       Munich
5     </span>
6     is the capital of Bavaria <br>
7     (population: <span property="dbo:population">12440000</span>).
8  </p>
```

Abbildung 7: Beispiel für Linked Data mit RDFa Markup

JSON-LD hingegen bringt Linked Data in die Welt von REST-Schnittstellen und NoSQL-Datenbanken. Die Notation wird dem Namen entsprechend im JSON-Format aufgeschrieben [19]:

```
1  {
2    "@context": {
3      "dbo": "http://dbpedia.org/ontology#"
4    },
5    "@id": "http://dbpedia.org/resource/Bavaria",
6    "dbo:capital": {
7      "@id": "http://dbpedia.org/resource/Munich"
8    },
9    "dbo:population": "12440000"
10 }
```

Abbildung 8: Beispiel für Linked Data in JSON-LD Notation

Die Daten sind auf diese Art nun im Web miteinander vernetzt, sie sind allerdings erst dann wirklich nützlich, wenn man die Informationen auch wieder extrahieren kann. Zu diesem Zweck hat das W3C 2008 die „SPARQL Query Language" [20] als Abfragesprache RDF standardisiert. Mit Hilfe dieser SQL-ähnlichen Syntax können Anfragen an öffentlich RDF-

Endpunkte wie z.B. die von *dbpedia.org*. Eine Abfrage nach allen Hauptstädten deutscher Bundesländer sieht beispielweise wie in Abbildung 9: SPARQL Abfrage aus.

http://dbpedia.org/sparql

```
1  PREFIX dct: <http://purl.org/dc/terms/>
2  PREFIX dbc: <http://dbpedia.org/resource/Category:>
3  SELECT * WHERE {
4    ?city dct:subject dbc:German_state_capitals
5  }
```

Abbildung 9: SPARQL Abfrage

SPARQL kann in diversen Tools sowie speziellen Browsern für Linked Data verwendet werden.

Wie die Welt der verknüpften Daten als Graph visualisiert aussehen kann zeigt Abbildung 10: State of the LOD Cloud (2014), lod-cloud.net

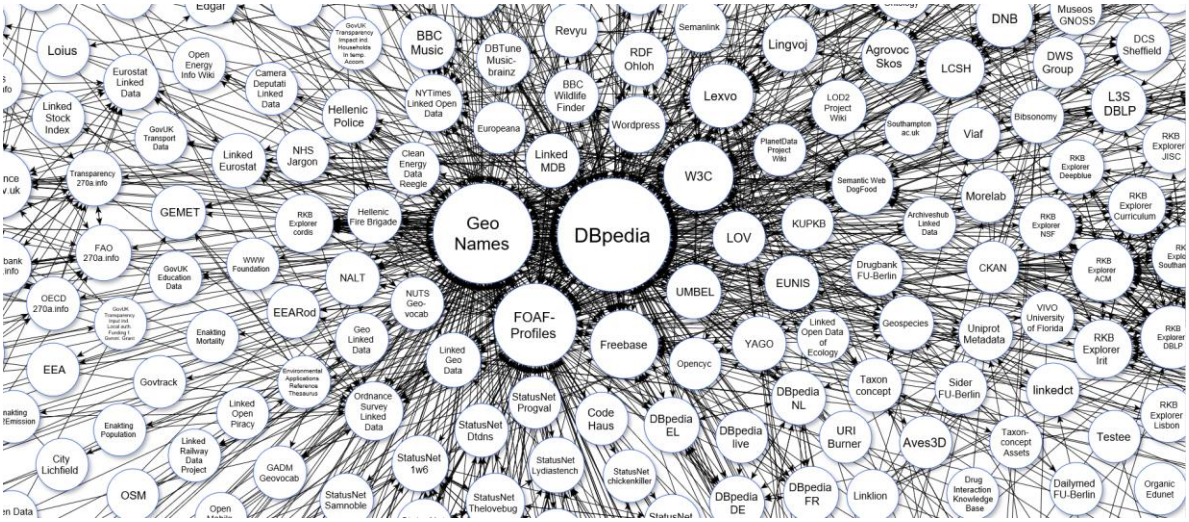

Abbildung 10: State of the LOD Cloud (2014), lod-cloud.net [21]

2.4 Small Data & Big Data

Der Begriff ***Small Data*** hat sich im Grunde erst mit dem Aufkommen von *Big Data* entwickelt. Er beschreibt letztendlich all das, was an Daten zuvor schon existierte und was nicht den weiter unten beschriebenen Charakteristiken für *Big Data* entspricht. Eine exakte, allgemeingültige

Definition gibt es allerdings nicht. jwork.org [22] bezeichnet Daten als „small“, wenn diese noch für einen Menschen verständlich sind. Die Handhabbarkeit für ein Individuum schließt auch Allen Bonde mit seinem Versuch einer allgemeinen Definition ein:

> *„Small data connects people with timely, meaningful insights (derived from big data and/or "local" sources), organized and packaged – often visually – to be accessible, understandable, and actionable for everyday tasks."* [23]

Als Beispiele für Small Data werden häufig „Excel-Tabellen“ genannt, als Grenze der für Menschen zumutbaren Datenmenge und -vielfalt. Dies zeigt, dass der Fokus hier in der Regel auf strukturierten Daten liegt, was jedoch semistrukturierte und unstrukturierte Daten (Webseiten, Fotos, E-Mails, Fließtext etc.) zu Unrecht außenvorlässt, obwohl diese sehr wohl in den selben Bereich fallen, angenommen man unterscheidet nur zwischen Small und Big Data.

Auch für ***Big Data*** gibt es bislang keine allgemeingültige Definition. Oftmals wird damit nur die reine Menge an Daten in Verbindung gebracht, was allerdings deutlich zu kurz gegriffen ist, wie nachfolgend gezeigt wird.

Big Data wurde anfangs vor allem durch die **3Vs** *Volume* (Volumen), *Velocity* (Geschwindigkeit) und *Variety* (Vielfalt) charakterisiert. Mangels entsprechender Datenbanktechnologien konnten zunächst allerdings immer nur zwei dieser drei Eigenschaften für Daten erreicht werden. Durch die stetig steigende Rechenleistung von Computern und das Aufkommen neuer Technologien wie NoSQL-Datenbanken (siehe Abschnitt 4.4) wurde diese Einschränkung jedoch durchbrochen. Zusätzliche Charakteristiken für Big Data wurden identifiziert. Die wichtigsten allgemein anerkannten Eigenschaften werden nachfolgend kurz erläutert:

Volumen

Der Begriff **Big** Data impliziert naturgemäß zunächst das reine Volumen an sich, also die Größe der Daten im Sinne von benötigtem Festplattenspeicher. Dabei gibt es keine festen Grenzen, ab welcher Menge genau Daten dann *Big* sind. Dies würde in Anbetracht des rasanten Wachstums an weltweit produzierten und gespeicherten Daten auch keinen Sinn ergeben. Während eine Privatperson mittlerweile Festplatten mit mehreren Terrabytes (2^{40} Bytes) kaufen kann, schätzte Thomas Barnett Jr. [24] den jährlichen Internet-Traffic in 2015 bereits auf über ein Zettabyte (2^{70} Bytes, also 1 Milliarde Terrabytes).

Allein Facebook [25] generierte und speicherte in 2014 täglich rund 600 TB an Daten, bei Twitter sind es in 2016 rund 500 Millionen Tweets pro Tag [26], bei denen zu jedem Tweet 33 verschiedene Metainformationen gespeichert werden ([2], S.71). Der *Large Hadron Collider* am CERN registriert gar 600 Millionen mal pro Sekunde Kollisionsereignisse, deren Messwerte jeweils etwa 1 MB groß sind. Da diese Datenmengen für eine dauerhafte Speicherung bei weitem zu groß wäre, müssen diese zunächst durch spezielle Algorithmen auf ca. 1 GB pro Sekunde reduziert werden. [27]

Geschwindigkeit

Nicht nur die rein existierende Menge an Daten, sondern auch die Geschwindigkeit, in denen diese produziert und verarbeitet werden, ist charakteristisch für *Big Data*. Während beispielsweise in Studien Daten vergleichsweise langsam, etwa in Befragungen oder Experimenten und oftmals manuell gesammelt werden, ist die Produktion und Verarbeitung bei *Big Data* meist kontinuierlich und oftmals in (harter und/oder weicher) Echtzeit. Hierdurch steigt das Datenvolumen drastisch und die Infrastruktur muss auf einen stetigen Datenfluss ausgelegt sein ([2] S.76f).

Vielfalt

Gerade die bei Big Data zusammen vorkommenden, verknüpften und vermischten Arten von Daten (siehe Abschnitt 2.1) wie unstrukturierten und (semi-)strukturierten Daten z.B. bestehend aus Text, Fotos, Videos, Audio in verschiedensten Dateiformaten bilden eine große Herausforderung für Administratoren und Entwickler. Die 600 TB Daten pro Tag bei Facebook bestehen beispielweise zu einem großen Teil aus von Nutzern veröffentlichten Beiträgen, die aus Text und Metadaten bestehen und oftmals mit Fotos, Videos oder Webseiten sowie Kommentaren, Likes und Benutzerprofilen verknüpft sind ([2] S.77).

Umfang

In vielen Small Data-Studien versucht man, anhand repräsentativer Daten, die z.B. durch die Befragung entsprechend ausgewählter Personen gewonnen wurden, Schlussfolgerungen zu ziehen. Bei Big Data hingegen wird versucht, möglichst alle verfügbaren Daten zu sammeln und auszuwerten, anstatt sich auf Stichproben zu beschränken. Dies wird vor allem durch den technischen Fortschritt im Sammeln von Daten verbunden mit einer „mehr-ist-besser"-Mentalität (größere Repräsentativität für valide Analysen) begünstigt. Am Beispiel Facebook

ist dies ebenfalls zu sehen, da hier neben Posts, Likes, besuchten Seiten und Profilen sowie verlinkten Webseiten und Interessen auch Werte wie Aktivitäts- und Abwesenheitsdauer mitgeloggt werden ([2] S.72f).

Detailgrad und Indexikalität

Neben dem Umfang steigt auch die Qualität beziehungsweise der Detailgrad einzelner Werte oder Datenblöcke. Beispielsweise werden die Satellitenaufnahmen auf Google Maps laufend durch höher aufgelöste Fotos ersetzt, die einen noch besseren Blick auf die Erde ermöglichen. In anderen Forschungsbereichen lassen sich zudem durch verbesserte Technik und Messgeräte noch mehr Werte pro Sekunde mit noch genaueren Messwerten ermitteln (vgl. CERN im Abschnitt **Volumen**).

Zu dem erhöhten Detailgrad kommt hinzu, dass vermehrt mithilfe von eindeutigen Kennzeichnern alle Bereiche der digitalen und physischen Welt indexierbar gemacht werden. Fotos oder sonstige digitale Produkte können beispielweise durch ein spezielles Wasserzeichen mit einem Käufer in Verbindung gebracht werden, um eventuell eine unrechtmäßige Nutzung Dritter nachvollziehen zu können.

Ein anderes Beispiel sind Barcodes im Supermarkt, über die eine bestimmte Produktsorte gekennzeichnet wird, bei denen einzelne Produkte der gleichen Art jedoch nicht unterschieden werden können. Durch den zunehmenden Gebrauch von RFID-Chips ist es nicht mehr nur möglich zu bestimmen z.B. welche Sorte Milch gekauft wurde, sondern es lässt sich anhand der einmaligen ID im Chip und einfacher Radio-Transponder der gesamte Weg dieser einen bestimmten Packung von der Molkerei über die Lagerung und den Verkauf voraussichtlich in Zukunft bis hin zum smarten Kühlschrank in der eigenen Wohnung nachvollziehen ([2] S.73f).

Relationalität

Der Wert von Datensätzen steigt im gleichen Maße, in denen diese mit anderen Daten kombiniert werden und daraus neue Erkenntnisse gewonnen werden können. Obwohl in Big Data-Szenarien selten relationale Datenbanken verwendet werden, sind die gespeicherten Datenblöcke doch in der Regel stark untereinander vernetzt und ermöglichen es, aus diesen Verbindungen durch entsprechende Analysen Erkenntnisse zu gewinnen. Hier ist man durch die flexiblen Systeme zur Datenspeicherung nicht auf ausschließlich strukturierte Daten

beschränkt, sondern kann Verbindungen zwischen allen Arten von Daten darstellen und nutzen ([2], S. 74ff).

Flexibilität

Eine weitere Eigenschaft von Big Data ist die strukturelle Flexibilität der Datensätze, die es ermöglicht, auch Änderungen während des laufenden Sammelprozesses an der Datenstruktur durchzuführen – beispielsweise um neue Metriken bei der Betrachtung von Userinteraktionen mit einer Webseite mitaufzunehmen. Hier wäre es bei herkömmlichen relationalen Datenbanksystemen notwendig, zunächst den Sammelprozess zu stoppen und aufwändig das Datenschema zu ändern. Flexibilität bedeutet zudem, dass die verwendeten Systeme darauf ausgelegt sind, für entsprechend große Datenvolumen (horizontal) skalieren zu können (siehe auch Abschnitt 4.4, NoSQL) ([2], S.77f).

Vergleich Small Data und Big Data

Durch den starken Trend hin zu riesigen Datenmengen wird oftmals der Wert von Small Data unterschätzt. In Big Data-Szenarien werden in der Regel Daten verarbeitet, die leicht in großen Mengen gesammelt werden können, wie beispielsweise Nutzerinteraktionen mit einer Website oder einem Programm, Messwerte von dauerhaft betriebenen Sensoren oder anderweitig aufgefangene Daten die als Nebenprodukt abfallen. Diese Daten müssen anschließend oft mit großem technischen und wissenschaftlichem Aufwand (z.B. über statistische Modelle etc.) verarbeitet und analysiert werden, um tatsächlich wertvolle Ergebnisse zu erhalten. Man könnte also sagen, Big Data muss erst auf Small Data heruntergebrochen werden, um für Menschen verständlich und nützlich zu sein.

Durch die lange Historie der Arbeit mit bezüglich des Umfangs beschränkten Datensätzen ist es dagegen möglich, auf vielfach wissenschaftlich erprobte Methoden der Datenerfassung, Verarbeitung und Analyse zurückzugreifen. Die Art der Datenerhebung ist zudem besonders für Außenstehende leichter nachvollziehbar und auf möglich Probleme oder Irrtümer hin überprüfbar. Zudem können Daten erhoben werden, die kaum automatisch gesammelt werden können, wie dies beispielsweise bei persönlichen Meinungen durch Umfragen getan werden kann ([3] S.29). In Anbetracht der Tatsache, dass Big Data enorme technische Herausforderungen mit sich bringt und damit im Bereich von Open Data bislang kaum eine

Rolle spielt, werden Daten im weiteren Verlauf dieser Arbeit im Kontext von Small Data betrachtet, sofern dies nicht explizit anderweitig angegeben ist.

3. Open Data

3.1 „Open“-Bewegungen

Der Begriff *Open* findet sich im Namen zahlreicher Interessensgruppierungen wieder, sei in es in *Open Source*, *Open Government* oder eben *Open Data*. Diese ähneln sich in der Regel nicht nur namentlich, sondern meist auch in ihrer Überzeugung und Motivation, aus denen sich deren Mitglieder für die entsprechenden Themen einsetzen, wobei sich die einzelnen Definitionen bezüglich der Offenheit durchaus in gewissen Details unterscheiden.

Deshalb folgt zunächst ein kurzer Blick auf ein paar der bekanntesten Vertreter der „Open“-Bewegungen bevor das Thema *Open Data* näher beleuchtet wird.

Open Source

Der Begriff *Open Source* wurde im Februar 1998 von der kurz zuvor gegründeten *Open Source Initiative* (OSI) formalisiert. Auslöser hierfür war die Veröffentlichung des Netscape Source Codes und der Wunsch der Gruppierung, eine Alternative zum politisch und philosophisch geprägten *Free Software* Label zu schaffen [28].

Entsprechend wurde auch eine Open Source Definition formuliert (ursprünglich basierend auf den Debian Free Software Guidelines), denen Open Source Projekte beziehungsweise die hierfür gewählten Lizenzen entsprechen müssen [29]:

- **Weitergabe:** Die Software darf uneingeschränkt und ohne die Zahlung von Lizenzkosten weiterverbreitet beziehungsweise als Teil eines Softwarepakets auch weiterverkauft werden, egal ob als Quellcode oder in kompilierter Form. Wird der Source Code nicht mit weitergegeben, muss dokumentiert werden, wo und wie dieser erhältlich ist.
- **Source Code:** Das Projekt muss den Source Code im ursprünglichen Zustand (nicht vorverarbeitet oder verschlüsselt) enthalten und frei zugänglich sein.
- **Lizenz:** Es muss erlaubt sein, Änderungen an der Software durchzuführen und abgewandelte Werke unter denselben Lizenzbestimmungen weiterzuverbreiten. Zudem darf die Lizenz weder gegen Gruppen oder Personen noch gegen Einsatzgebiete (egal ob privat oder kommerziell, Bereiche wie z.B. Genforschung) diskriminieren. Weiter muss die Lizenz unabhängig von einem etwaigen Produkt, in

dem der Code verwendet wird, gültig sein und darf auch nicht andere, damit gebündelte Software, einschränken.

Während vor allem zu Beginn die Open Source-Bewegung maßgeblich von Einzelpersonen und Non-Profit-Organisationen geprägt wurden, entdecken vermehrt auch große Softwarefirmen die Vorzüge von Open Source (z.B. Google Android, Microsoft .NET Framework).

Unter dem Begriff Open Source versteht man mittlerweile nicht nur mehr Softwareprojekte, sondern auch Sub-Bewegungen wie *Open Source Hardware* und *Open Source Architecture*.

Open Government

Ziel von Open Government ist es, das Handeln von Regierungen und Verwaltungen möglichst offen zu gestalten, um mehr Transparenz gegenüber der Bevölkerung und der Wirtschaft und damit zum einen die Möglichkeit der Kontrolle von staatlichen Organen aber auch eine erhöhte Teilhabe und verbesserte Zusammenarbeit zu erlangen [30].

Im Zentrum von Open Government steht meist die Forderung nach der Öffentlichmachung von Daten (*Open **Government** Data*), bezieht sich allerdings unter anderem auch auf Aspekte wie die Verwendung von Open Source Software oder die Förderung von Open Innovation („*Lösung gesellschaftlicher Herausforderungen durch Staat und Gesellschaft*“ [31]).

Open Access

Die Open Access Bewegung setzt sich dafür ein, dass wissenschaftliche Literatur kostenlos und öffentlich, für jedermann zugänglich im Internet zur Verfügung steht. Interessierte sollen Volltexte *„lesen, herunterladen, kopieren, verteilen, drucken, in ihnen suchen, auf sie verweisen und sie auch sonst auf jede denkbare legale Weise benutzen können, ohne finanzielle, gesetzliche oder technische Barrieren […]“* [32] können sollen. Hierbei geht es vor allem darum, wissenschaftliche Veröffentlichungen unter entsprechenden Open-Access-Lizenzen zu fördern, die oft im Konflikt beispielsweise mit der Vergabe von exklusiven Rechten an Verlage stehen.

Argumente für Open Access [33] beinhalten unter anderem, dass Ergebnisse öffentlich subventionierter Studien auch wieder der Öffentlichkeit zugänglich gemacht werden sollten. Zudem können offen zugängliche Dokumente von mehr Personen gelesen, kontrolliert und

zitiert werden – es profitiert also sowohl die Sichtbarkeit als auch die Qualität hiervon. Die Verfügbarkeit über das Internet – eine der Kernforderungen von Open Access – sorgt gerade im Vergleich zur herkömmlichen Veröffentlichung in einer wissenschaftlichen (Closed-Access-) Zeitschrift für ein deutlich höheres Verbreitungspotenzial durch den weltweit möglichen Zugriff, verstärkt durch die Auffindbarkeit in Suchmaschinen und Nachweisdiensten. Dies fördert auch die internationale und interdisziplinäre Zusammenarbeit und erhöht die Forschungseffizienz. Allerdings bleiben hier alle Urheber- und Verwertungsrechte beim Autor und ist damit etwas restriktiver als etwa bei Open Source oder Open Data.

Open Education

Die *Open Education* Bewegung hat sich als Ziel gesetzt, Bildung im Allgemeinen und die damit verbundenen Aspekte entsprechend den *Open*-Prinzipien kostenlos und uneingeschränkt, also in erster Linie im Internet, zur Verfügung zu stellen. Hierzu zählt der Aufbau, die Unterstützung und Verbreitung entsprechender Angebote, beispielsweise kostenloser Lehr- und Lernmaterialien sowie Online-Kurse, offen zugänglicher Fachbücher und Medien, freier Bibliotheken etc.

Neben den genannten Open-Bewegungen entstehen laufend neue Interessensgruppierungen, die sich unter dem *Open*-Label für den freien Zugang zu Informationen in einem bestimmten Themengebiet einsetzen, z.B. Open Design & Hardware, Open Spending, Open Bibliography oder Open Research.

3.2 Definition: Open Data

Während man zunächst die Definition von *Open Data* als relativ auf der Hand liegend einschätzen könnte, kann vor allem zu Missverständnissen führen auf welche Aspekte der Daten sich die „Offenheit“ konkret bezieht – beispielsweise auf den Zugang, die Verwendung und Abwandlung oder der Verkauf der Daten. Ein Datensatz kann also durchaus frei zugänglich sein, allerdings nicht frei verwendbar.

Einige Institutionen haben sich deshalb das Ziel gesetzt, die Idee der idealen „Offenen Daten“ genauer zu definieren. Federführend ist hier die Open Knowledge Foundation mit der Open Definition (siehe Tabelle 1), deren erster Entwurf aus dem Jahre 2005 mittlerweile mehrfach überarbeitet wurde und aktuell in Version 2.1 vorliegt. Diese weithin anerkannte Sammlung

von Regeln sieht für offene Daten vor, dass diese kostenlose zugänglich sein sollten, weiterverarbeitet und -verteilt werden dürfen sowie explizit zu kommerziellen Zwecken verwendet werden können. Dabei kann der Ersteller der Daten verlangen, dass die Daten nur unter Nennung der ursprünglichen Quelle (*Attribution*, also die Zuschreibung) und unter derselben oder einer vergleichbaren Lizenz weitergegeben werden können.

Tabelle 1 – Definition von Open Data nach Open Definition

Lizenz	Das zu veröffentlichende Werk (Daten etc.) muss unter einer kompatiblen Lizenz (siehe Abschnitt Lizenzen) veröffentlicht werden. Diese darf nicht durch zusätzliches, anderweitige Klauseln (z.B. Rechtsansprüche, Patente) eingeschränkt werden. Die Lizenz muss stets dem Werk angehängt sein und ist unwiderruflich.
Weitergabe	Das Werk darf zudem uneingeschränkt unter denselben Bedingungen im Original oder in bearbeiteter Fassung weiterverbreitet werden.
Verwendung	Die Lizenz muss unwiderruflich das Recht auf kostenlose Verwendung und (auch teilweise) Weiterverarbeitung der Daten garantieren. Hierbei darf kein Verwendungszweck untersagt werden.
Technische Voraussetzungen	Das Werk muss in maschinell lesbarer und einfach bearbeitbarer Form zur Verfügung stehen. Das Datenformat muss dabei ohne Einschränkungen mit kostenloser / Open-Source-Software zu verarbeiten sein (siehe Abschnitt Datenformate).
Kompensation	Für den Zugang zum Werk dürfen kein Geld oder anderweitige Gegenleistungen verlangt werden.
Zugang	Das Werk sollte vollständig kostenlos online für jedermann zugänglich sein.
Akzeptable Einschränkungen	Die Lizenz kann die Nennung der Quelle sowie der Lizenz selbst fordern, wenn Daten verwendet oder weitergegeben werden.

	Zudem kann verlangt werden, dass abgewandelte Werke als solche kenntlich gemacht werden, z.B. durch eine Namensänderung oder eine neue Versionsnummer. Die Lizenz kann außerdem vorschreiben, dass weiterverbreitete Versionen eines Werkes ohne zusätzliche technische Hürden versehen werden, die die gegebenen lizenzrechtlichen Vorgaben einschränken würden.

Zusammenfassung basierend auf [34]

Aufbauend auf den in der Open Definition festgehaltenen Grundlagen formte eine aus 30 Open Government-Befürwortern [35] bestehende Arbeitsgruppe in den Vereinigten Staaten die *8 Prinicples of Open Government Data* [36]. Diese behandeln neben den Gesichtspunkten der Verwendung und der Verbreitung von Werken auch die Art und Herkunft der Daten sowie ihrer Qualität. Zudem sind die Regeln, wie aus dem Titel bereits hervorgeht, in erster Linie als Leitfaden für Regierungsorganisationen gedacht. Die meisten Punkte sind allerdings auch für nichtstaatliche Herausgeber von Daten relevant.

Tabelle 2: 8 Prinicples of Open Government Data

1. Vollständigkeit Alle verfügbaren Daten sollen vollständig veröffentlicht werden und als Gesamtpaket zugänglich sein. Ausgenommen sind Daten, die aus Datenschutz-, Sicherheits-, oder rechtlichen Gründen nicht veröffentlich werden können.
2. Aus erster Hand Daten sollten mit höchstmöglichem Detailgrad gesammelt und vollständig sowie unbearbeitet veröffentlicht werden. Bei aggregierten und verarbeiteten Datensätzen muss zusätzlich das unbearbeitete Quellmaterial zugänglich gemacht werden.
3. Zeitnahe Veröffentlichung Daten sind nach ihrer Erhebung so schnell wie möglich zu veröffentlichen, um deren Wert zu erhalten.

4. Zugänglichkeit

Daten müssen der größtmöglichen Zahl von Nutzern für die größtmögliche Zahl von Verwendungszwecken zugänglich gemacht werden.

Hierzu müssen die Daten über das Internet in einer Art und Weise zur Verfügung stehen, in der sie für so viele Nutzer wie möglich verwendbar sind, unabhängig von etwaigen Behinderungen oder eingesetzter Software beziehungsweise Hardware.

Daten müssen zudem über Protokolle und in Formaten zugänglich sein, die den aktuellen Industriestandards entsprechen. Sollten diese Standards die Verbreitung und breite Verwendung der Daten behindern, sollten zudem alternative Protokolle und Formate verwendet werden.

5. Maschinelle Lesbarkeit

Daten müssen maschinell lesbar und einfach zu verarbeiten sein. Hierfür müssen die Datensätze folgende Eigenschaften besitzen:

- Sinnvolle Struktur
- Angemessene Zeichencodierung
- Kein Fließtext als Ersatz für tabellarische, normierte Datensätze
- Keine Bilder von Text als Ersatz für den Text selbst

6. Nicht diskriminierend

Alle Daten müssen für jeden anonym (auch über Proxys) und ohne Registrierung zugänglich sein.

7. Nicht proprietär

Daten müssen in Formaten zur Verfügung stehen, die nicht unter der vollständigen Kontrolle einer einzelnen Organisation stehen. Hiermit soll ermöglicht werden, dass Daten unabhängig von Firmenentscheidungen bleiben und stets ohne kostenpflichtige Software verarbeitet werden können und auch in Zukunft nicht unlesbar werden.

Für den Fall, dass die Verwendung eines nicht-proprietären Standards dazu führt, dass weniger Personen die Daten nutzen können, sollten verschiedene Formate zum Einsatz kommen.

8. Lizenzfrei

Daten dürfen keinen Copyright-, Patent- oder Markenschutzansprüchen sowie Betriebsgeheimnissen unterliegen. Ausgenommen sind hier Daten, bei denen berechtigte Datenschutz-, Sicherheits- und rechtliche Bedenken bestehen.

Gerade im Regierungsumfeld muss klar gekennzeichnet werden, welche Daten öffentlich zugänglich und ohne rechtliche Einschränkung nutzbar sind.

Überprüfbarkeit der Einhaltung

Als Zusatz zu den acht aufgestellten Regeln muss sichergestellt werden können, dass ein Datensatz diese auch einhält. Hierzu ist es notwendig, eine für den Datensatz verantwortliche Person festzulegen, die im Falle von Fragen oder Problemen mit den Daten oder bei Verstößen gegen die festgelegte Lizenz kontaktiert werden kann.
Zudem muss ein weiteres administratives oder juristisches Organ die Einhaltung dieser Prinzipien überwachen können.

Zusammenfassung basierend auf "The Annotated 8 Principles" of Open Government Data (übersetzt) [37]

Aufbauend auf diesen Regeln gibt es weitere Versuche, Open Data möglichst umfassend zu definieren, so zum Beispiel von der *Sunlight Foundation* mit den 31-Punkte umfassenden *Open Data Policy Guidelines* [38], die festlegen, welche Daten veröffentlich werden sollten, wie diese öffentlich gemacht werden können und wie sichergestellt werden kann, dass die Open-Data-Regeln eingehalten werden. Im Kern berufen sich diese weiteren Regularien allerdings stets auf die oben genannten Punkte.

Für den Erfinder des World Wide Webs Tim Berners-Lee sind Daten hingegen erst dann vollkommen offen, wenn diese zusätzlich zu den genannten Definitionen in Form von Linked Data (siehe 2.3) vorliegen. Hierzu hat er eine 5-Sterne-Bewertungsskala entworfen, anhand derer die *Offenheit* von Daten gemessen werden kann. Abhängig von der angestrebten Bewertung ist jede Stufe mit gewissen Kosten und Nutzen verbunden. Beispielsweise bedeutet eine verbesserte Zugänglichkeit der Daten oftmals einen deutlich höheren Wartungsaufwand,

etwa um alle Daten in RDF-Form zu veröffentlichen und Links zu anderen Datensätzen zu pflegen.

★	Stelle deine Daten im Web unter einer offenen Lizenz bereit. Das Format ist dabei egal
★★	Stelle Daten in einem strukturierten Format bereit (z. B. Excel anstelle eines eingescannten Bildes einer Tabelle)
★★★	Verwende offene, nicht proprietäre Formate (z. B. CSV statt Excel)
★★★★	Verwende URIs um Dinge zu bezeichnen, damit deine Daten verlinkt werden können
★★★★★	Verlinke deine Daten mit anderen Daten um Kontexte herzustellen

3: 5-Sterne-Modell für Offene Daten [39]

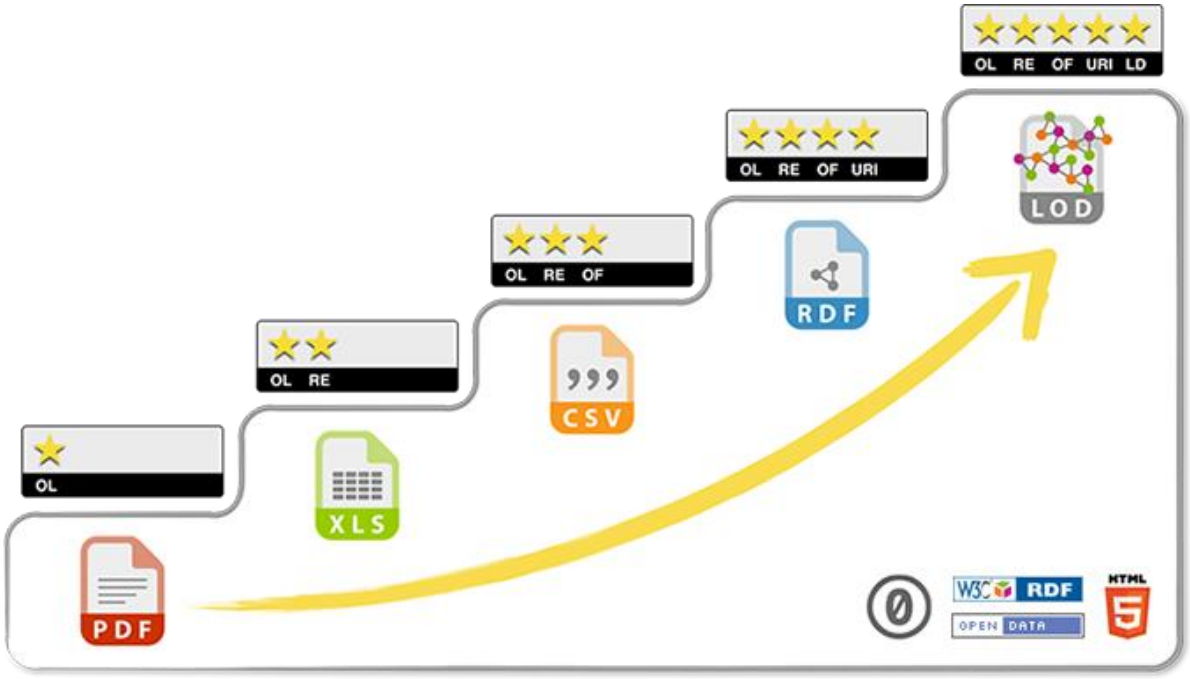

Abbildung 11: 5-Sterne-Modell für Offene Daten [39]

Open Data-Projekte können zudem durch entsprechende Zertifikate die Einhaltung der erläuterten Definitionen und Regeln demonstrieren. Diese werden beispielsweise vom Open

Data Institute [40] ausgestellt. Anhand eines Fragebogens wird bewertet, wie gut Datenpakete dem Open-Data-Gedanken entsprechen. Entsprechend gibt es Zertifikate in den Varianten Bronze, Silber, Gold und Platin, die die Qualität der Daten sowie die Infrastruktur, in denen diese angeboten werden, bewerten.

Zusammenfassend lässt sich sagen, dass trotz der recht detaillierten Definitionen Open Data ein sehr breit gefächerter und vielfältig verwendeter Begriff ist, der sich nicht auf ein bestimmtes Thema beschränkt. Zahlreiche Open Data Initiativen wurden in den letzten Jahren gestartet, besonders im Bereich des Open Government. Entsprechend sind all die Daten in ihrer Entstehung und Verbreitung subventioniert, ein nachhaltiges Konzept hierfür wurde bislang noch nicht gefunden. Ob und wie das Thema sich auch in der freien Wirtschaft weiterentwickeln wird, bleibt demnach abzuwarten, zumal die potenziellen Nachteile von Open Data bislang kaum erforscht wurden ([2] S.66).

3.3 Lizenzen

Die wichtigste Grundlage für jedes Open Data-Projekt ist die Wahl einer geeigneten Lizenz. Hierfür muss sich der Ersteller der Daten im Klaren sein, welche Rechte er abgeben möchte – und gegebenenfalls welche Rechte er überhaupt abgeben kann. Lizenzen, die den Open-Data-Anforderungen gemäß der Open Definition (siehe Tabelle 1) entsprechen, müssen alle folgende Rechte einräumen:

Jeder Nutzer darf unwiderruflich die Daten **bearbeiten**, **weitergeben**, und **kommerziell verwenden**.

Zu den nicht Open Data kompatiblen Lizenzen gehörten beispielsweise alle Creative-Commons-Varianten, die eine (teilweise) Weiterverbreitung und Verarbeitung sowie eine kommerzielle Verwendung untersagen. [42]

Tabelle 4"Open Data"-konforme Lizenzen

Lizenz	Bereich	Attribution	Share-alike
Creative Commons CCZero (CC0)	Inhalte, Daten	Nein	Nein
Open Data Commons Public Domain Dedication and Licence (PDDL)	Daten	Nein	Nein
Creative Commons Attribution 4.0 (CC-BY-4.0)	Inhalte, Daten	Ja	Nein
Open Data Commons Attribution License (ODC-BY)	Daten	Ja	Nein
Creative Commons Attribution Share-Alike 4.0 (CC-BY-SA-4.0)	Content, Data	Ja	Ja
Open Data Commons Open Database License (ODbL)	Data	Ja	Ja

Quelle: [41]

3.4 Vorteile von Open Data

Die Gründe der Open Data-Befürworter, warum Daten für alle kostenlos zugänglich gemacht werden sollten, sind vielfältig. Viele Argumente beziehen sich konkret auf die Veröffentlichung von Daten durch Staaten und Regierungen. Die nachfolgend genannten Gründe gelten allerdings größtenteils auch für andere potenzielle Anbieter von offenen Daten, wie Firmen, Nichtregierungsorganisationen oder auch Privatpersonen.

Als fundamentales Argument wird oft genannt, dass Daten der Menschheit gehören [43]. Dies bezieht sich in erster Linie auf grundlegende wissenschaftliche Erkenntnisse, die aus dieser Sicht allen Menschen zugänglich sein sollten, sodass auch jeder einzelne sowie die Gemeinschaft hiervon profitieren kann. Prominente Beispiele sind Forschungsfelder wie die DNA-Sequenzierung (Genome), Erkenntnisse im medizinischen Bereich oder Umweltdaten, etwa zur globalen Erwärmung.

Befürworter berufen sich hier vor allem auf die Tatsache, dass Fakten nicht vom Urheberrecht geschützt werden können und somit jedem zugänglich sein sollten. Hier sind die Grenzen zwischen ungeschützten Faktenzusammenstellungen und wissenschaftlichen Werken, welche sehr wohl dem Urheberrecht unterliegen, fließend und können oftmals erst gerichtlich geklärt werden [44].

Ein weiterer Kerngedanke in der Debatte um Open Data ist, dass vor allem Informationen, deren Entstehung durch Steuergelder finanziert wurde, der Allgemeinheit zugänglich sein sollten. Dies betrifft sowohl Regierungen und Universitäten (deren Daten in der Regel ohnehin öffentlich gemacht werden) als auch Firmen und sonstige Einrichtungen, deren Forschungsarbeit und Studien durch öffentliche Gelder finanziert werden.

Zusätzlich wird hierdurch eine unabhängige Kontrolle öffentlicher Organe bezüglich der Verwendung solcher Gelder durch die Öffentlichkeit ermöglicht. Entscheidungsprozesse innerhalb der Organisationen werden damit einsehbar und Erfolg für die Öffentlichkeit messbar gemacht. Gerade in einer Zeit, in der viel über die Verschwendung öffentlicher Gelder gesprochen wird, können so auch Verantwortlichkeiten geklärt werden [45].

Auch intern erhalten verschiedene Abteilungen Einblick in die Arbeit ihrer Kollegen, können diese selbst weiterverwenden und bei Bedarf Feedback geben. Zudem können diese die Vorteile der Transparenz direkt erleben und möglicherweise dazu ermuntert werden, selbst ihre Daten offenzulegen. Dies kann das Image der Organisation verbessern, da man zeigt, dass sie Öffentlichkeitsarbeit ernst nimmt. Sie kommt mit der Bevölkerung beziehungsweise den Kunden in Kontakt und verstärkt die Kommunikation mit diesen. Dies fördert Vertrauen und steigert das Ansehen nach außen [46].

Durch die freie Verfügbarkeit der Daten im Internet müssen zudem weniger Behördenanfragen bearbeitet werden, was den hierfür verantwortlichen Abteilungen der Behörden Arbeitszeit einspart. Zudem kann die Öffentlichkeit bei der Pflege der Daten mithelfen und auf eventuelle Probleme oder Fehler hinweisen und auf den Daten basierende Analysen wieder mit den Behörden teilen. All dies spart am Ende Steuergelder, da diese Arbeit kostenlos von engagierten Bürgern durchgeführt werden könnte.

Weiter haben die Daten im Besitz der öffentlichen Hand oftmals auch großen finanziellen Wert. Eine Veröffentlichung kann sich positiv auf die Wirtschaft auswirken. Neue Dienstleistungen und Anwendungen können hierdurch entstehen, da durch die Daten z.B.

Bedürfnisse oder Zusammenhänge ersichtlich werden und sowohl für etablierte als auch neue Unternehmen das Treffen von informierten Entscheidungen und die Erschließung neuer Märkte ermöglicht werden [47]. Hier ist hervorzuheben, dass diese Daten anderweitig oft nur teuer gekauft werden können und so vielen Personen und Organisationen nicht zu Forschungs- und Innovationszwecken zur Verfügung stehen würden. Zudem können die möglicherweise teuer gesammelten Daten vielfach wiederverwendet werden, was die ursprünglichen Investitionen relativiert.

Außerdem spricht für die kostenlose Weitergabe der Daten, dass hierdurch auch Dienstleistungen entstehen können, die nicht nur wirtschaftlichen, sondern auch gesellschaftlichen Mehrwert liefern. Beispielsweise könnten für die Bürger interessante Informationen verständlich in Form von Webseiten oder Apps aufgearbeitet werden.

Unabhängig davon ermöglicht der Zugriff auf diese Art von Informationen, dass Bürger ihre Umgebung besser kennen lernen und Zusammenhänge besser verstehen können, sehen wie Steuergelder verwendet werden und letztendlich mehr Vertrauen aufbauen. Vor allem die Veröffentlichung soziokulturell relevanter Daten kann dazu führen, dass sich Privatpersonen über die Arbeit mit diesen verstärkt gesellschaftlich und politisch engagieren [48].

Die genannten Argumente entspringen zwar vieler unterschiedlicher Blickrichtungen, bezogen auf diverse Bereiche und mit durchaus unterschiedlichen – meist wirtschaftlichen oder politisch motivierten – „Hintergedanken“, haben sich jedoch in der Vergangenheit besonders in der Politik als äußerst wirksam gezeigt. Von den 28 Mitgliedsländern der Europäischen Union plus den vier Ländern der Europäischen Freihandelsassoziation betreiben 87% bereits ein eigenes, zentrales Datenportal [49]. Ein sehr anschauliches Beispiel, wie Bürger durch entsprechenden Zugang zu Informationen Einfluss nehmen können kommt aus den Vereinigten Staaten von Amerika, dem Vorreiter im Bereich Open Data. Hier konnte ein privater Analyst anhand von öffentlich zugänglichen Daten zu verteilten Strafzetteln für Falschparker in New York nachweisen, dass zahlreiche vermeintliche Vergehen fälschlicherweise bestraft und Bußgelder von mehreren $100.000 zu unrecht gezahlt wurden [50].

Im privaten Sektor ist der Open-Data-Gedanke bisher noch nicht in diesem Maße als Chance erkannt worden, wie dies beispielsweise bereits bei Open-Source-Projekten der Fall ist, obwohl viele der hier erkannten Chancen wie der Unterstützung von Kunden durch Analyse

und Meldung von Fehlern oder eine bessere Vernetzung mit der *Community* auch auf den Bereich von Open Data zutreffen können.

3.5 Bedenken gegen Open Data

Es gibt jedoch auch Gegenstimmen sowohl aus politischen als auch wirtschaftlichen Lagern, die der pauschalen Veröffentlichung aller in Frage kommenden Daten ablehnend gegenüberstehen. Auch hier sind entsprechende Gegenargumente verschiedenen Perspektiven zuzuordnen.

Zunächst ist die Bereitstellung von Open Data mit nicht unerheblichem Arbeitsaufwand verbunden, dessen unmittelbarer Nutzen, vor allem die Vorteile für die Organisation selbst, nicht direkt sichtbar sind. Somit ist es vorstellbar, dass diese Bestrebungen z.B. innerhalb einer Firma oder Behörde möglicherweise gegenüber einem Vorgesetzten schwierig zu rechtfertigen sind [51].

Doch auch aus wirtschaftlichen Interessen lehnen manche Gegner vor allem die kostenlose Zurverfügungstellung von Daten ab. Sie argumentieren damit, dass die Regierung in direkte Konkurrenz zu Firmen tritt, die solche Daten teils teuer produzieren oder einkaufen bzw. damit handeln. In dem Moment, wo Zugang und Nutzung durch Dritte (egal ob Regierungs- oder sonstige Organisationen bzw. Privatpersonen) kostenfrei ermöglicht wird, könne diese Investitionen auf einen Schlag wertlos sein.

Als Beispiel hierfür eignet sich der Disput um die Tagesschau-App (auch wenn die Inhalte nicht den oben genannten Definitionen von Open Data entsprechen). Hiergegen hatten einige private Verlage rechtliche Schritte eingeleitet, da aus ihrer Sicht dieser zusätzliche Service in dieser Form „presseähnlich“ sei und damit in direkter Konkurrenz zu den eigenen Online-Angeboten stehe. Die Komplexität des Problems lässt sich allein daran festmachen, dass in drei gerichtlichen Instanzen drei unterschiedliche Urteile gesprochen wurden [52].

Oft scheitert die Veröffentlichung von Daten an diesen einfachen wirtschaftlichen Interessen. Gerade im privaten Sektor möchten die Urheber in der Regel ihre Werke und Ideen marken- und patentrechtlich schützen und einen Gewinn aus der geleisteten Arbeit ziehen – vor allem, wenn sie selbst in finanzielle Vorleistung hierfür gegangen sind ([2] S57f).

Aus politscher Sicht besteht die Angst, sich und seine Arbeit durch die Veröffentlichung von Daten angreifbar zu machen. Diese könnten durch unabsichtlich oder von politischen Gegnern

durch gezielte Fehlinterpretationen und Missdeutungen gegen die Regierung verwendet werden. Zudem besteht die Gefahr, dass versehentlich gemachte Fehler als mutwillige Manipulation der Daten z.B. zur Legitimierung eines politischen Kurses gesehen werden könnten. Die durch Open Data geschaffenen, zusätzlichen Kontrollmöglichkeiten der Arbeit in Regierungen und Verwaltungen durch die Öffentlichkeit kann ebenso dazu führen, dass betroffene Beamten diese Entwicklung kritisch sehen.

[31] [53]

Doch auch abseits wirtschaftlicher und politischer Interessen gibt es Bedenken, die auch von Open-Data-Befürwortern getragen werden und die in erster Linie rechtlicher Natur sind. So ist es beispielsweise mit geringem Aufwand möglich, offene Datensäte abzuändern und weiterzuverbreiten, um gezielt Falschinformationen zu streuen. Schon die Quelldaten können versehentlich oder bewusst verfälscht, erfunden oder unvollständig sein, um eigene Interessen wie den Beleg einer Studie zu stützen und daraus wirtschaftliche oder politische Konsequenzen zu ziehen [54].

Andere Forscher befürchten, dass gerade finanziell stark aufgestellte Firmen und Interessensverbände die veröffentlichten Daten ausgiebig analysieren und gerade bei wissenschaftlich ungewissen Themen diese zu ihren Gunsten auslegen können und damit Gesetzgebungen zu beeinflussen, wie dies beispielsweise in den USA durch die Tabakindustrie bzw. -lobby geschehen ist [54] [55]. Privatpersonen und gemeinnützigen Organisationen fehlt hierfür oft der finanzielle Hintergrund.

Der Missbrauch von Open Data zum Schaden anderer ist eine weitere Gefahr. So können beispielsweise offengelegte Sicherheitslücken in Software und IT-Systemen schnell zweckentfremdet werden. Wissenschaftler aus den Niederlanden und den USA haben in der Vergangenheit z.B. ihre Forschungen an genmanipulierten Varianten der Vogelgrippe trotz potenziell hilfreicher Ergebnisse für deren Bekämpfung unterbrochen, aus Angst, die damit verbundenen Publikationen der Resultate und Vorgehensweisen könnte als Anleitung zur Nachahmung missbraucht werden [56].

Auch bei auf den ersten Blick unbedenklichen Daten kann die Gefahr bestehen, dass sie ungewollt sensible Informationen offenbaren. So wurden im Jahr 2014 von der Stadt New York Daten über Taxis veröffentlicht, die über 173 Millionen einzelne Fahrten beinhalteten, inklusive GPS-Informationen der Start- und Endpunkte sowie den Fahrtkosten. Hierin waren

zudem sensible Daten wie die Taxi- und Lizenznummern der Fahrer enthalten, die allerdings verschlüsselt und damit auf den ersten Blick anonymisiert waren. Jedoch gelang es dem Informatiker Vijay Pandurangan, den verwendeten Verschlüsselungsalgorithmus zu identifizieren und in wenigen Stunden Arbeit die entsprechenden Identifikationsnummern dank deren fixem Format wieder zu entschlüsseln. Dies ermöglichte in Kombination mit weiteren frei zugänglichen Daten die Identifikation einzelner Fahrer und damit die Möglichkeit, exakte Bewegungsprofile sowie Informationen über deren Einnahmen zu extrahieren [57].

Vor allem in Deutschland ist der Datenschutz ein sehr hoch angesehenes Gut. Jeder Bürger hat das Grundrecht auf informationelle Selbstbestimmung. Dies bedeutet, dass jede Person ein Recht darauf hat, selbst zu bestimmen, welche persönlichen Informationen wo und in welcher Weise veröffentlicht werden. Es ist also zunächst eine Einwilligung notwendig, die zudem jederzeit widerrufen werden kann. Dies steht jedoch im Kontrast zu der Verpflichtung des Staates auf Transparenz, der z.B. durch Open Government-Initiativen und in Form von Informationsfreiheitsgesetzen nachgekommen wird. Das genannte Taxi-Beispiel zeigt deutlich die Schwierigkeit, die Grenze zwischen diesen beiden gesetzlich geregelten Aspekten zu finden [58].

Die genannten Bedenken werfen auch die Frage nach der Haftung auf, sollte die Verwendung von Open Data zu einem Schadenfall führen. Als Anbieter von Daten kann man hier unter Umständen mit Rechtsansprüchen konfrontiert werden, wenngleich in Deutschland zumindest die kostenfreie Weitergabe von Open Source-Software als Schenkung angesehen wird und man hier nur für „*arglistig verschwiegene Mängel*“ haften muss [59].

3.6 Open Data Tools

Rund um das Thema Open Data hat sich mit der Zeit ein Ökosystem aus Informationsquellen und Anleitungen, Werkzeugen zur Erfassung, Pflege und Weiterverbreitung von Daten sowie zentrale Anlaufstellen für die Suche nach offenen Datensätzen gebildet. All diese versuchen, ausgewählte Aspekte im Open Data Lifecycle zu vereinfachen.

Ein umfassender Überblick würde den Rahmen dieser Arbeit sprengen, deshalb werden nachfolgend drei Tools vorgestellt, die sich in der Community etabliert haben und die sich im Hinblick auf Einsatzzweck und gebotenen Funktionalitäten mit dem in Kapitel 4 und 5 vorgestellten Konzept und dem zugehörigen Prototyp vergleichen lassen. Ein spezieller Fokus

liegt hierbei auf der Einfachheit der Handhabung und der damit verbundenen Eignung, technisch weniger versierten Menschen die Arbeit mit Open Data zu vereinfachen.

CKAN

CKAN ist eine Datenmanagementplattform, die von der *Open Knowledge Foundation* entwickelt wird und kostenlos zur Verfügung steht. Das Projekt ist Open Source, damit kann CKAN auch selbst gehostet oder als Service gemietet werden. Mit *datahub.io* steht zudem eine Instanz der Software der Öffentlichkeit zur Verfügung, über die jedermann selbst Daten finden, herunterladen oder veröffentlichen kann.

Innerhalb von CKAN stehen zahlreiche Funktionen zur Verfügung, die das Veröffentlichen von Daten und die anschließende Verwaltung und Kommunikation bezüglich dieser vereinfachen. Datensätze können über ein einfaches Webformular unabhängig vom verwendeten Format hochgeladen oder verlinkt werden. Zusätzlich können Meta-Informationen wie z.B. Beschreibung, Schlagworte, Autor, Version oder auch benutzerdefinierte Felder ausgefüllt werden, um die Daten besser auffindbar zu machen. Zudem lässt sich direkt eine passende Lizenz aus einer vorgegebenen Liste auswählen.

Nachdem die Daten veröffentlicht wurden, können diese über eine eingebaute Kommentarfunktion mit anderen Nutzern diskutiert werden. Der Aktivitäten-Feed liefert einen Überblick darüber, wann die Daten zuletzt aktualisiert und welche sonstigen Änderungen am Datensatz vorgenommen wurden.

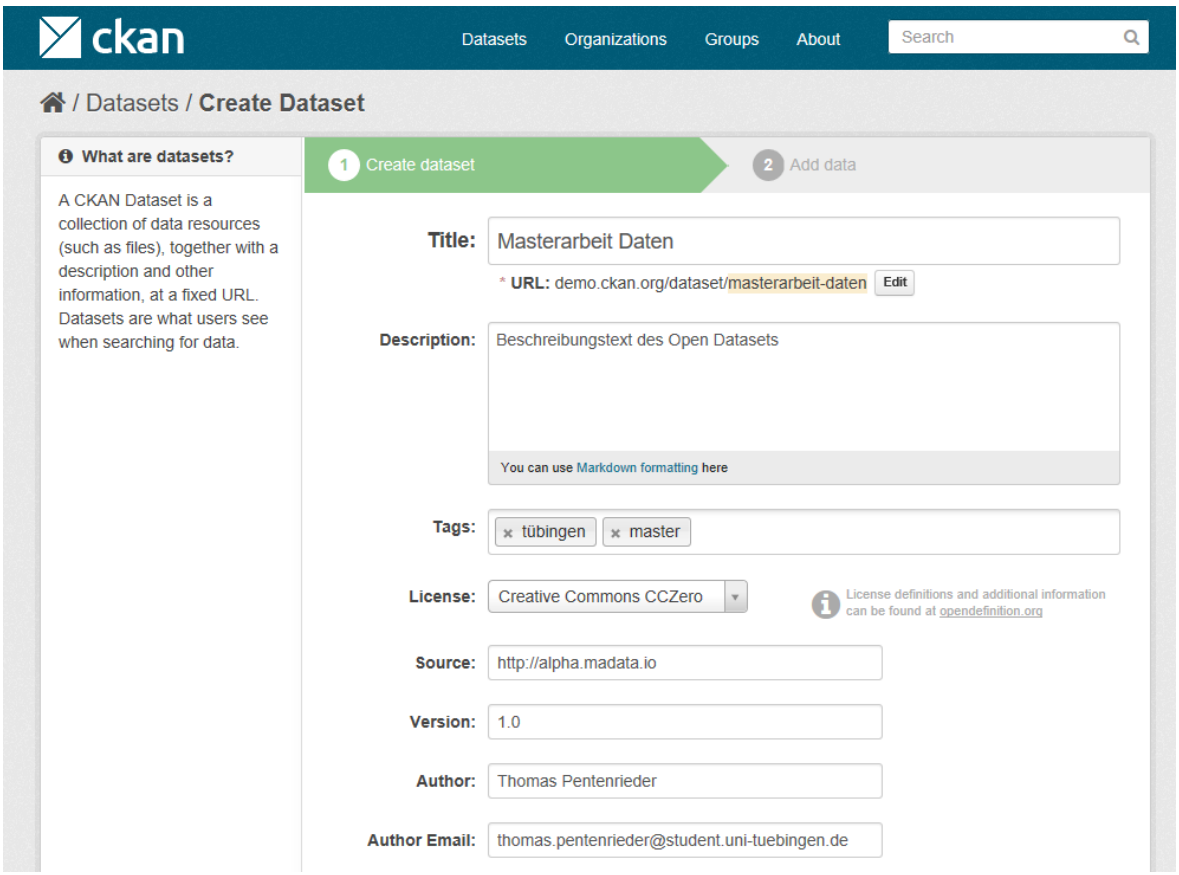

Abbildung 12: Screenshot von CKAN Bearbeitung eines Datensatzes

Daten, die in tabellarischer Form (z.B. als CSV oder Excel-Datei) vorliegen, können direkt in CKAN automatisch aufgelistet und in Form von Graphen oder auf Landkarten visualisiert werden. Ein weiterer Pluspunkt ist die JSON-API, über die Datensätze abgefragt und aktualisiert werden können. Zudem ist das System um weitere Funktionen erweiterbar, die bei Bedarf auch selbst entwickelt werden können.

Der größte Vorteil der Plattform ist, dass diese vor allem von nicht-technisch versierten Personen genutzt werden kann, da der Fokus hier offensichtlich auf der einfachen Benutzbarkeit des Systems liegt. Alle nötigen Funktionen sind einfach über die Weboberfläche zu bedienen. Lediglich das Installieren eigener Instanzen ist technisch aufwändig und nicht trivial.

Ein großer Nachteil von CKAN ist, dass Nutzer keine Daten ändern bzw. konkrete Änderungsvorschläge unterbreiten können. So muss der Besitzer des Datensatzes Änderungen stets selbst manuell einpflegen und anschließend das gesamte Datenpaket erneut hochladen.

Das Tool ist dementsprechend dann geeignet, wenn das Sammeln der Daten bereits abgeschlossen ist [60].

GitHub

GitHub.com ist in erster Linie ein auf der Versionsverwaltungssoftware *Git* basierender Service zur Verwaltung von Softwareprojekten. Besonders beliebt ist der Dienst bei Open Source-Projekten, da diese GitHub kostenlos verwenden können.

Open Data-Projekte haben oftmals sehr ähnliche technische Bedürfnisse wie Open Source Projekte, weshalb gerade Softwareentwickler oft dieselben Tools für beide Bereiche verwenden. GitHub bietet hier neben kostenlosem Speicherplatz zahlreiche Funktionen, die die Verwaltung von Daten vereinfachen.

Die eigentlich für Code gedachte Versionskontrolle eignet sich gut, um Änderungen in Datensätzen nachvollziehen und ältere Versionen einsehen zu können. Zudem hat jeder Nutzer die Möglichkeit, sich in Form eines *Forks* eine eigene Kopie der Daten zu erstellen und diese nach Belieben weiterzubearbeiten. Die bearbeiteten beziehungsweise korrigierten Daten können dann auch wieder in den Quelldatensatz übernommen werden. Zudem können über *Issues* die Daten und etwaige Probleme diskutiert werden.

Das Hauptproblem bei GitHub ist die hohe Einstiegshürde, vor allem für nicht-technische Nutzer. Überall werden Fachtermini aus der Softwareentwicklung verwendet, was viele Interessenten schnell abschrecken kann. Um Daten beispielsweise innerhalb einer Software (wie z.B. Excel) korrigieren zu können, muss zunächst ein Fork erstellt und die Daten *ausgecheckt* und bearbeitet werden. Anschließend müssen diese Änderungen *commited* und *gepushed* werden. Zuletzt wird ein *pull request* erstellt, damit diese Änderungen wieder vom Besitzer des Projekts übernommen werden können. Daten können zwar auch über einen Webeditor bearbeitet werden, hier allerdings immer nur in ihrer bloßen Textform, ohne dass Eingaben inhaltlich oder strukturell geprüft werden. Dass dies jedoch in der Praxis exakt so praktiziert wird, zeigt den Bedarf an anderen beziehungsweise zusätzlichen Lösungen.

Es gibt Bestrebungen, GitHub auch für technisch nicht versierte Nutzer durch zusätzliche Werkzeuge einfacher zugänglich zu machen. Beispielsweise ermöglicht Octopub.io, über ein simples Webinterface den Upload von bestehenden Datensätzen zu GitHub zu vereinfachen und zusätzliche Meta-Informationen wie Beschreibung und Lizenz hinzuzufügen. Die

nachträgliche Bearbeitung der so veröffentlichten Daten wird hierdurch allerdings auch nicht einfacher gestaltet. Außerdem erfüllen die auf diese Art angebotenen Datensätze nicht die Anforderungen an Linked Data, wie sie von Tim Berners-Lee gefordert werden (siehe Abschnitt 2.3 & 3.2), da einzelne Einträge nicht direkt verlinkt werden können, sondern immer nur die gesamte Sammlung.

Wikidata

Wikidata.org ist das Open Linked Data Portal der *Wikimedia Foundation*. Dieses wird analog zu Wikipedia von freiwilligen Helfern gepflegt und überwacht. Hier entscheidet also das Kollektiv (bzw. ein *ranghohes Mitglied*), welche Daten ins Konzept passen und welche nicht.

Zudem werden keine abgeschlossenen Datensätze gesammelt, sondern einzelne Informationselemente (sogenannte *Items*) wie Themen, Konzepte, Personen oder Objekte. Beispielsweise sind „Film“, „Liebe“, „Edward Snowden“ und „Smartphone“ jeweils ein *Item*, welches auch nur ein einziges Mal in der Datenbank existiert. All diese einzelnen Datenelemente sind bei Wikidata sind untereinander verlinkt und damit Teil eines riesigen Graphen. Hierdurch ist es sehr einfach möglich, verknüpfte Informationen zu einem bestimmten Stichwort zu erhalten. Dies wird beispielsweise auf Wikipedia eingesetzt, um neben Artikeln die Infobox zu befüllen. Demenstprechend ist Wikidata nicht für in sich abgeschlossene Datensätze gedacht, die nicht einzeln identifizierbare Items enthalten, wie beispielsweise Temperaturverläufe an einem bestimmten Ort [61].

Wappen	Deutschlandkarte

Basisdaten	
Bundesland:	Baden-Württemberg
Regierungsbezirk:	Tübingen
Landkreis:	Tübingen
Höhe:	341 m ü. NHN
Fläche:	108,12 km²
Einwohner:	87.464 *(31. Dez. 2015)*[1]
Bevölkerungsdichte:	809 Einwohner je km²
Postleitzahlen:	72070, 72072, 72074, 72076
Vorwahlen:	07071, 07073, 07472
Kfz-Kennzeichen:	TÜ
Gemeindeschlüssel:	08 4 16 041
Stadtgliederung:	23 Stadtteile
Adresse der Stadtverwaltung:	Am Markt 1 72070 Tübingen
Webpräsenz:	www.tuebingen.de
Oberbürgermeister:	Boris Palmer (Bündnis 90/Die Grünen)

Abbildung 13 Infobox auf Wikipedia, Daten aus Wikidata [79]

Basierend auf den Einblicken in die verfügbaren Werkzeuge und den genannten Defiziten hinsichtlich des Funktionsumfangs und der Nutzerfreundlichkeit wird im Folgenden ein Konzept vorgeschlagen, welches versucht, für diese Mängel eine Lösung zu bieten.

4. Projekt Konzeption

Aus der Analyse der bestehenden Open Data-Werkzeuge in Kapitel 3 wurde der Bedarf für ein neues Tool erkannt, welches die oben identifizierten, fehlenden Funktionalitäten bietet. Nachfolgend wird unter Einbeziehung der Erkenntnisse aus Kapitel 2 und 3 ein Konzept für eine Open Data-Plattform vorgeschlagen, welches die Aspekte der Datenverwaltung und Veröffentlichung inkorporiert.

4.1 Ziel des Projektes

Ziel des Projektes ist, eine Plattform zu schaffen, über die gemeinschaftlich an Open Data-Projekten gearbeitet werden kann. Es sollen also nach den oben erläuterten Open Data-Richtlinien Daten mithilfe der Öffentlichkeit gesammelt werden können, die einer zuvor vom Initiator definierten Datenstruktur genügen. Hierbei steht nicht nur das Hinzufügen neuer Daten im Fokus, sondern auch das Korrigieren bestehender Einträge. Die hieraus entstehenden Datenprojekte stehen dementsprechend auch der Öffentlichkeit gemäß den Open Data Prinzipien zur Verfügung. Im Kern entspricht die Idee der einer Versionsverwaltung, allerdings nicht für Quellcode (z.B. GitHub, Team Foundation Server) oder Fließtexte (z.B. Content Management Systeme wie WordPress), sondern für strukturierte Daten.

4.2 Beispiel für ein Open Data-Projekt

Im Straßenverkehr trifft man immer wieder auf LKWs, die auf ihrer Rückseite eine orange Tafel mit Zahlenkombinationen angebracht haben. Diese Zahlen geben Auskunft darüber, welcher Gefahrstoff gerade transportiert wird. Dies ist ein wichtiger Hinweis vor allem für Rettungskräfte, sollte der Transporter in einen Unfall verwickelt sein. Die UN-Nummern, die den Stoff codieren, wurden von der United Nations Economic Commission for Europe (UNECE) festgelegt und werden in unregelmäßigen Abständen im Rahmen von Revisionen aktualisiert. Die überarbeitete Fassung (aktuell Revision 19 aus dem Jahr 2015) wird dann als PDF mit mehreren 100 Seiten veröffentlicht. Um die hier enthaltenen Daten also weiter verarbeiten zu können, müssen diese erst aufwändig aus dem PDF extrahiert und in ein brauchbares Datenformat konvertiert werden [62].

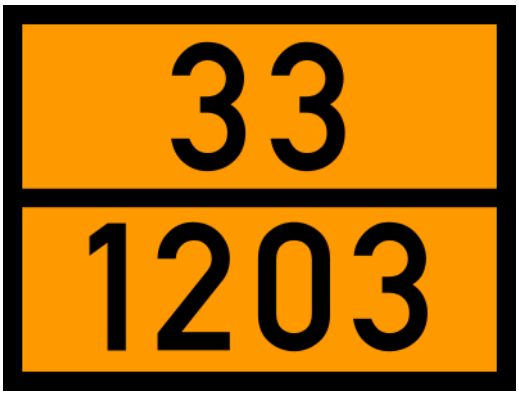

Abbildung 14: Gefahrentafel mit Gefahrnummer (oben) und UN-Nummer (unten) [78]

Eine vollständige Liste der UN-Nummern, die den Bedingungen von Open Data vor allem bezüglich der einfachen maschinellen Verarbeitung z.B. durch die Bereitstellung als JSON oder XML entsprechen würden, sind, trotz der eigentlich offenen Verfügbarkeit der Informationen, im World Wide Web nicht auffindbar. Die zugänglichste Version ist eine tabellarische Auflistung auf Wikipedia, die allerdings auch hier nur als HTML-Dokument vorliegt und dementsprechend nur mit beträchtlichem Aufwand weiterverarbeitet werden kann.

Hier würde dementsprechend ein Dienst von Nutzen sein, über den diese Daten für Mensch (Website) und Maschine (REST API) lesbar sind und Fehler bzw. Änderungen aus neuen Revisionen direkt von betroffenen Nutzern (z.B. von Berufskraftfahrern, die eine mobile App basierend auf den Daten verwenden) vorgenommen werden können.

4.3 Herausforderungen & Lösungsansätze

Einfachheit

In allen gängigen Definitionen von Open Data wird großer Wert darauf gelegt, dass die Daten ohne jegliche Einschränkung universell zugänglich sein sollen. Dies ist bezüglich der verwendeten Werkzeuge oftmals nicht der Fall, denn diese haben häufig eine sehr hohe technische Einstiegshürde, wenn diese z.B. selbst gehostet werden müssen oder für technisch versierte Menschen konzipiert wurden (GitHub). Ziel ist es deshalb, dass jeder mit grundlegenden *Internet-Kenntnissen* den Prototyp verwenden kann.

Lösungsansatz: Um das Werkzeug möglichst einfach verwendbar zu machen, wurde der Prototyp als Webanwendung konzipiert, sodass lediglich ein herkömmlicher Browser und eine Internetverbindung zur Bedienung notwendig sind. Um eigene Datenstrukturen zu definieren kommt ein einfach zu bedienender, visueller Formulareditor zum Einsatz. So muss der Nutzer weder Datenbankkenntnisse mitbringen noch sich direkt mit Schema-Notationen (siehe JSON Schema, Abschnitt 4.4) befassen.

Flexible Datenhaltung

Jeder Datensatz hat andere Erfordernisse bezüglich der enthaltenen Datenstruktur. Da diese vom Nutzer festgelegt werden können soll, ist es nicht möglich, bereits im Vorfeld bei der Anwendungsentwicklung ein entsprechendes Datenbankschema zu definieren. Hier muss eine

flexible Lösung gefunden werden, die es ermöglicht, erst zur Laufzeit entsprechende Anforderungen an das Format der Daten zu formulieren.

Lösungsansatz: Die Speicherung der Daten erfolgt in Form von semistrukturierten JSON-Dokumenten in einer NoSQL-Dokumentendatenbank. Um sicher zu gehen, dass diese stets die gewünschte Struktur besitzen, werden neue und geänderte Dokumente anhand des Anfangs definierten JSON Schemas validiert.

Berechtigungen

Hierfür bedarf es ein sinnvolles Berechtigungskonzept, um die Mitarbeit vieler Menschen an Datenprojekten zu fördern und gleichzeitig eine hohe Qualität der Beiträge zu gewährleisten. Hierzu gehört auch, Vorkehrungen gegen Daten-Vandalismus, also dem absichtlichen verschlechtern der Datenqualität durch das Löschen korrekter oder dem Hinzufügen falscher Informationen zu treffen.

Lösungsansatz: Standardmäßig ist es nur dem Ersteller eines Datensatzes möglich, Veränderungen (hinzufügen, editieren, löschen) am Datenstamm vorzunehmen. Anderen Benutzern ist es jedoch möglich, Änderungsvorschläge einzusenden, die vom Ersteller gesichtet und je nach Qualität akzeptiert, weiter editiert oder abgelehnt und damit gelöscht werden können. Um die Administration eines Datensatzes zu erleichtern, können zudem weitere Benutzer mit Moderationsrechten ausgestattet werden.

Sicherheit

Immer wenn Nutzer Daten vor allem über Formulare an einen Server senden und diese später auf der Webseite auch zu sehen sein sollen, ist dies mit nicht zu vernachlässigenden Sicherheitsproblemen verbunden. Abgesehen von z.T. absichtlich falschen Daten könnte ein Angreifer versuchen, beispielsweise JavaScript Code einzuschleusen (sogenanntes Cross-Site-Scripting), der bei anderen Nutzern dann ausgeführt wird, sobald diese den infizierten Datensatz ansehen. Weitere potenzielle Angriffspunkte existieren in Form der Datenbanken – sowohl relationale als auch NoSQL – die Ziele für entsprechende (No)SQL-Injections werden könnten. Diese Angriffe können dazu führen, dass z.B. persönliche Daten eines Nutzers ohne sein Wissen an Dritte weitergeleitet werden oder es zum Datenverlust auf der Plattform kommen kann.

Lösungsansatz: Für die Anwendung und deren Nutzer möglicherweise schädliche Eingaben müssen abhängig vom potenziellen Angriffsziel separat behandelt werden. Hierzu müssen kritische Sonderzeichen (vor allem Anführungsstriche, Klammern etc.) entsprechend durch sogenannte *Escape characters* maskiert werden, damit diese nicht fälschlicherweise von Server, Browser oder Datenbank als Befehlszeichen interpretiert werden.

4.4 Verwendete Technologien

Im Nachfolgenden werden die für den Prototyp essenziellen Technologien und Konzepte erläutert. Dabei liegt der Fokus auf der Datenhaltung. Wie diese in die Architektur der Anwendung passen, zeigt Abbildung 17: Architektur des Prototyps (Technologien).

NoSQL (not only SQL) wird als Sammelbegriff für Datenbanken verwendet, die nicht dem klassischen relationalen Stil entsprechen. Sie besitzen meist die nachfolgenden Eigenschaften, allerdings nicht zwangsläufig alle hiervon [63]:

- **Schemalos**
  Dies ist der offensichtlichste Unterschied zu relationalen Datenbanksystemen (RDS), welche stets normalisierte Daten (also in tabellarischer, untereinander verknüpfter Form) in einem im Vorfeld festgelegten Schema speichern. NoSQL Datenbanken können hingegen Daten in unterschiedlichen Strukturen speichern.
- **Horizontale Skalierung**
  Ein Großteil der NoSQL Datenbanksysteme wurde von Anfang an explizit so konzipiert, dass diese horizontal skalieren, also durch das Hinzufügen von zusätzlichen Rechner/Knoten (*scale out*) erweitert werden können. Relationale Datenbanken sind dagegen meist darauf ausgelegt, auf einer Maschine optimal zu laufen, um vor allem den ACID-Prinzipien zu genügen.
- **BASE statt ACID**
  Während bei relationalen Datenbanken großer Wert auf die ACID-Prinzipien (*Atomicity*, *Consistency*, *Isolation* und *Durability*, zu Deutsch *Atomarität*, *Konsistenz*, *Isolation* und *Dauerhaftigkeit*) gelegt werden, ist bei verteilten (NoSQL-) Systemen vor allem die ständige Konsistenz der Daten problematisch, da Änderungen an Daten stets erst an mehrere Rechner weitergegeben werden müssen. Entsprechend dem CAP-Theorem [64] ist es nicht möglich, „*gleichzeitig die drei Eigenschaften Konsistenz (C), Verfügbarkeit (A) und Partitionstoleranz (P) zu garantieren.*“ [65]

Stattdessen setzt man hier tendenziell auf die BASE-Prinzipien (Basically Available, Soft state, Eventual consistency). Hier wird in Kauf genommen, dass der Datenbestand für einen gewissen Zeitraum inkonsistent ist, bis alle Änderungen letztendlich auf alle Instanzen übertragen wurden (→ *eventual consistency*).

- **Große Datenmengen & einfache Replizierung**
  Durch die horizontale Skalierbarkeit von NoSQL Datenbanken ist es möglich, durch zusätzliche Rechner/Knoten nicht nur die Last auf das System zu verteilen, sondern auch hohe Speicherkapazitäten für große Datenmengen (*Big Data*, siehe Abschnitt 2.4) und deren Sicherungskopien sowie für Georedundanz zur Verfügung zu stellen.

Seit Anfang 2009 steigt die Zahl der NoSQL Datenbanksysteme rasant. Viele hiervon sind Open Source bzw. kostenlos und können selbst gehostet werden. Doch auch proprietäre Systeme beziehungsweise Dienste (beispielsweise von Microsoft, Google, Amazon) drängen verstärkt auf den Markt. Weit über 200 verschiedene Produkte zählen hier im weitesten Sinne dazu [63], die sich teils deutlich bezüglich der grundlegenden Herangehensweise an die Datenhaltung sowie Programmier- und Abfragesprachen, Einsatzgebiete, und sonstigen Funktionen unterscheiden.

Grundlegend sind vier Arten von NoSQL Datenbanksystemen bezüglich der Datenhaltung zu unterscheiden [66]:

1. **Key-Value Stores**
   Bei Key-Value Stores besteht jeder Datensatz aus einem oder mehreren Schlüssel-Wert Paaren, wie sie in diversen Programmiersprachen auch in Form von *assoziativen Arrays*, *Hashmaps* oder *Dictionarys* bekannt sind. Vereinfacht gleichen sie einer Tabelle, bei der jede Zeile eigene und beliebig viele unterschiedliche Spaltenüberschiften hat. Da anders als bei relationalen Datenbanktabellen keine leeren Zellen mit Platzhaltern belegt werden müssen, brauchen diese oftmals deutlich weniger Speicherplatz. Deshalb und aufgrund der guten Abfragegeschwindigkeit werden diese Art von Systemen oft für das In-Memory-Caching von Daten genutzt. Beispiele sind *Redis*, *Amazon DynamoDB* und *MemcacheDB*.
2. **(Wide) Column Family Stores**
   Column Family Stores (Spaltenbasierte Datenbanken) sind auf das spaltenweise Schreiben und Lesen von Daten (im Gegensatz zu zeilenweise wie bei RDS) optimiert.

Diese sind vor allem dann effizienter, wenn nur ausgewählte Spalten eines Datensatzes verarbeitet werden müssen, da hierbei nicht immer die ganze Zeile gelesen werden muss. Dies macht sich vor allem dann bemerkbar, wenn die unterschiedlichen Spalten auf verschiedenen physischen Rechnern gespeichert sind (horizontale Skalierung). Dementsprechend wird diese Art der Datenbank oftmals z.B. bei der Analyse von *Big Data* eingesetzt.

Beispiele sind *Apache Cassandra*, *Apache Hadoop / HBase* und *Amazon Redshift.*

3. **Document Databases**

   Dokumentendatenbanken werden in der Regel dazu verwendet, um teilweise strukturierte Daten (siehe Abschnitt 2.1) zu speichern. Diese liegen meist in JSON- oder XML-Form und können vom Schema her variieren.

   Beispiele sind *CouchDB*, *MongoDb* und *Microsoft DocumentDB.*

4. **Graph Databases**

   Graph-Datenbanken enthalten Knotenpunkte und (gerichtete) Kanten mit jeweils entsprechenden Properties. Sie können dazu verwendet werden, um beispielsweise Daten in RDF-Struktur (siehe Abschnitt 2.3) zu speichern.

   Beispiele sind *OrientDB*, *GraphDB* und *Titan*

Aufgrund ihrer Flexibilität bezüglich der Struktur eines Datensatzes könnte im Prototyp prinzipiell jede JSON-basierte Document Database verwendet werden. Dies ist wichtig, da die Datenstruktur der anzulegenden Datensätze erst zur Laufzeit bekannt ist.

JSON Schema

Um sicher zu gehen, dass die in der NoSQL-Datenbank gespeicherten JSON Dokumente strukturell und inhaltlich den Erwartungen des Datensatz-Erstellers entsprechen, werden diese mittels JSON Schema validiert. Hierbei handelt es sich um eine Schemasprache, die den Aufbau von JSON Daten beschreibt und selbst in validem JSON geschrieben wird. Derzeit liegt der Standard in der Version 4 bei der Internet Engineering Task Force als Entwurf vor. [67]

Die JSON Schema Spezifikation besteht aus drei Teilbereichen [68]:

1. **JSON Schema Core**

   JSON Schema Core beschreibt den grundlegenden Aufbau von JSON Schema zur

Beschreibung von JSON Dokumenten, beispielsweise Verwendung strukturellen Elementen (Array, Objekt) und Datentypen (Boolean, Integer, Number, Null, String).

2. **JSON Schema Validation**

   JSON Schema Validation enthält Schlüsselwörter zur Validierung von in JSON Dokumenten enthaltenen Daten, zum Beispiel maximale Zeichenlänge eines Strings, akzeptabler Wertebereich eines Integers oder komplexere Regeln in Form von regulären Ausdrücken (RegEx).

3. **JSON Hyper-Schema**

   JSON Hyper-Schema beschreibt Hyper-Media-Schlüsselwörter (z.B. Links zu anderen Dokumenten, Bilder in Base64 codiert etc.), innerhalb eines JSON Dokuments. Dies ist nützlich, um beispielsweise eine REST-API zu dokumentieren.

Da JSON Schema für JSON Dokumente sowohl strukturell als auch inhaltliche Validierungsregeln vorgeben kann, eignet sich dieser Standard hervorragend, um Nutzern die Eingabe von Daten, beispielsweise über ein entsprechend aufgebautes Formular, zu erleichtern. Zudem können die Daten hierüber sowohl Clientseitig (im Browser direkt bei der Eingabe) als auch Serverseitig (z.B. für Daten, die über die REST-API eingefügt werden sollen) validiert werden können. Diese Intention geht auch aus der Spezifikation hervor:

> *„another application may use a JSON Schema to build an interactive interface in order to collect user input according to constraints described by JSON Schema." [48], Abschnitt 4.1*

Beispiel

```
1  {
2    "name": "München",
3    "einwohner": 1388000,
4    "bundesland": "Bayern",
5    "sprachen": [
6      "Bayrisch",
7      "Deutsch"
8    ]
9  }
```

Abbildung 15: JSON Dokument entsprechend Abbildung 16

```
1  {
2      "$schema": "http://json-schema.org/draft-04/schema#",
3      "title": "Stadt",
4      "description": "Die besten deutschen Städte",
5      "type": "object",
6      "properties": {
7          "name": {
8              "title": "Name",
9              "description": "Name der Stadt",
10             "type": "string",
11             "validationMessage": "Geben Sie einen Namen ein"
12         },
13         "einwohner": {
14             "title": "Einwohnerzahl",
15             "type": "number",
16             "minimum": 0
17         },
18         "bundesland": {
19             "title": "Bundesland",
20             "type": "string",
21             "enum": ["Bayern", "Baden-Württemberg"]
22         },
23         "sprachen": {
24             "title": "Sprachen",
25             "type": "array",
26             "items": {
27                 "type": "string"
28             },
29             "minItems": 1,
30             "uniqueItems": true
31         }
32     },
33     "required": ["name", "einwohner"]
34 }
```

Abbildung 16: JSON Schema Beschreibung für JSON Dokument (Abbildung 15)

Prinzipiell wäre die Verwendung einer XML-basierten Dokumentendatenbank in Kombination mit Struktur- und Validierungsregeln in Form von XML Schemas möglich.

Programmierschnittstelle

Um die Vorgaben der Open Data-Richtlinien bezüglich der maschinellen Lesbarkeit der Daten zu gewährleisten, sind alle öffentlichen Datensätze nicht nur über ein Browserinterface, sondern auch über eine Programmierschnittstelle (*application programming interface*, kurz API) zugänglich. Diese basiert auf den Prinzipien *Representational State Transfer (REST)* Architektur, verbunden mit HTTP als Übertragungsprotokoll. Konkret bedeutet dies, dass die Abfragen über normale URLs im *http://*-Format erfolgen, wobei diese keine Methoden enthalten, sondern nur auf bestimmte Ressourcen (in diesem Fall ganze Datensätze oder einzelne Dokumente) zeigen [69].

Beispielsweise verweist die URL *http://alpha.madata.io/api/{**datasetid**}/items/{**itemid**}* auf ein bestimmtes Dokument mit der ID *itemid* innerhalb eines Datensatzes mit der ID *datasetid*. Durch die Verwendung von HTTP-Verben wird der Schnittstelle zusätzlich mitgeteilt, ob die Ressource einfach zurückgegeben (*GET*), neu angelegt (*POST*), überschrieben / modifiziert (*PUT*) oder gelöscht (*DELETE*) werden soll.

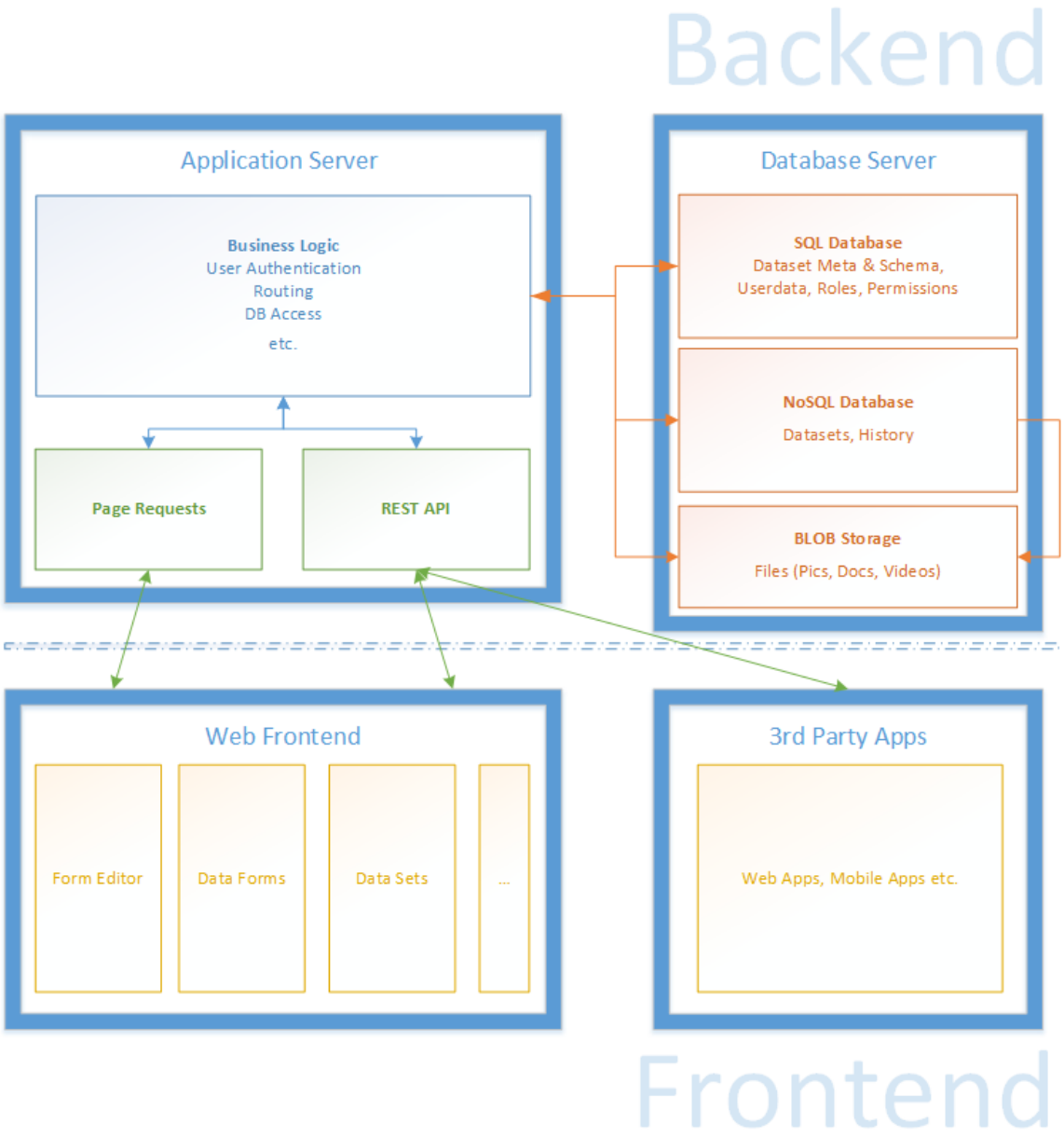

Abbildung 17: Architektur des Prototyps (Technologien)

5. Projekt Umsetzung

Nachfolgend wird die technische Umsetzung des in Kapitel 4 dargelegten Konzepts erläutert und dabei ein Überblick über die verwendeten Sprachen und Produkte sowie über den Funktionsumfang des Prototyps gegeben.

5.1 Eingesetzte Technologieprodukte

Das Konzept ist prinzipiell mit vielen verschiedenen konkreten Programmiersprachen, Plattformen und technischen Produkten umsetzbar. Die für diesen Prototyp getroffene Auswahl des Technologie-Stacks ist in erster Linie eine persönliche Präferenz. Die Gesamtarchitektur des Prototyps mit den verwendeten Produkten zeigt Abbildung 17: Architektur des Prototyps (Technologien).

Integrated Development Environment

Der gesamte Prototyp wurde innerhalb der integrierten Entwicklungsumgebung *Visual Studio 2015* (Update 3) entwickelt. Hierin wurden über die Paketmanager *nuget* (für server-seitigen Code, zum Beispiel *ASP.NET Core*, *Entity Framework*) und Bower (für client-seitigen Code, zum Beispiel *AngularJS* und Module, *Bootstrap*, *jQuery*) die benötigten *Dependencies* eingebunden.

Als Server wurde der Clouddienst Microsoft Azure gewählt, da hier die verwendeten, nachfolgend aufgelisteten Technologieprodukte mit geringem Konfigurationsaufwand an einem Ort gehostet und miteinander genutzt werden können. Zudem ist es möglich, direkt aus Visual Studio heraus das Projekt inklusive Datenbanken auf Azure zu veröffentlichen beziehungsweise zu aktualisieren.

ASP.NET Core

ASP.NET steht für Active Server Pages .NET und ist ein Web-Framework aus dem Hause Microsoft, welches mit dem Erscheinen des .NET Frameworks 2002 das zuvor entwickelte ASP als Nachfolger abgelöst hat. Es wird serverseitig ausgeführt und in C# dafür entwickelt und dabei das Model-View-Controller-Softwarearchitektur-Pattern verwendet [70].

Im Prototyp wird die neu entwickelte Version ASP.NET Core 1 (zuvor ASP.NET 5 genannt) in der zweiten Release Client Version (RC 2) verwendet. Diese ermöglicht es, Webanwendungen auf .NET Core aufbauend zu entwickeln, was die Anwendung erstmals unabhängig von lokalen .NET-Installationen und damit die Anwendung deutlich portabler

macht. Im Laufe der Entwicklungsarbeit wurde die finale Version von ASP.NET Core 1 veröffentlicht und das Projekt auf diese aktualisiert [71].

Datenbanken

Zur Speicherung der Kontodaten der Benutzer sowie den Meta-Informationen der Datensätze (Titel, Beschreibung, Schema etc.) wurde der in Azure gehostete relationale Datenbankservice *SQL database* verwendet.

Als Datenbank für die einzelnen, user-generierten Daten wurde DocumentDB eingesetzt. Der NoSQL-Dienst aus dem Hause Microsoft ist wie bereits aus dem Namen hervorgeht eine dokumentenorientierte Datenbank und nur in Azure erhältlich. Der NoSQL-Dienst kann auch nicht selbst gehostet werden. Dafür muss der Dienst nur minimal konfiguriert werden und enthält Funktionen wie die automatische Indexierung von Dokumenten (*Hash*- und *Range*, kann bei Bedarf angepasst werden), ein eigenes *REST-Interface* und *Stored Procedures* (in JavaScript), die den ACID-Anforderungen genügen. Außerdem kann DocumentDB beliebig horizontal skalieren, also die Dokumente auf mehrere Server verteilen und replizieren. Dies führt wie bei allen verteilten Systemen dazu, dass bei großer Last ein Kompromiss aus **Konsistenz** (Daten werden immer in der aktuellsten Version gelesen und bei einem Schreibvorgang synchron auf einen Großteil der Repliken übertragen) und **Verfügbarkeit** (Daten werden in akzeptabler Geschwindigkeit gelesen und geschrieben) eingegangen werden muss. Da die **Ausfallsicherheit** auf jeden Fall gegeben sein muss (vgl. CAP-Theorem, [72]), ist es bei DocumentDB möglich, in vier Stufen zwischen Konsistenz und Verfügbarkeit zu priorisieren [73]. Zudem ist für Anhänge (Bilder, Videos etc.) der Microsoft BLOB Storage direkt in DocumentDB integriert.

Application Programming Interface (API)

Die Programmierschnittstelle wurde über das ASP.NET Web API Framework realisiert, welches eine Erweiterung zum MVC-Framework darstellt. So können Anfragen mit entsprechenden HTTP-Verben (GET, POST, PUT, DELETE) versehen werden. Die Antworten hierzu werden standardmäßig im JSON-Format ausgegeben.

Um den Zugang zur API möglichst einfach zu gestalten wurde hierfür eine Definition entsprechend der *OpenAPI Specification*, besser bekannt unter dem bisherigen Namen *Swagger*, erstellt. Dies ermöglicht, anhand dieser Datei zum einen eine ausführliche API-

Dokumentation inklusive Testumgebung und zum anderen für externe Entwickler clientseitigen Code in allen gängigen Sprachen zur Verwendung der API zu generieren [74]. Streng genommen erfüllt die Schnittstelle nicht vollends alle Anforderungen an eine REST-Architektur, da in den Rückgabewerten keine weiterführenden Hyperlinks zu verknüpften Ressourcen enthalten sind. Somit ist man auf die Dokumentation der Schnittstelle angewiesen, was zwar wartungsaufwändiger ist (Dokumentation muss bei jeder Änderung angepasst werden), in der Praxis allerdings für Entwickler den Überblick verbessert und damit den Einstieg erleichtert.

Frontend

Beim Frontend der Webapplikation kommt neben klassischem HTML und CSS (Bootstrap) das von Google entwickelte JavaScript-Framework AngularJS sowie jQuery als Grundlage einiger Plug-Ins zum Einsatz.

Zur Anzeige der Daten wird jQuery-Plug-In DataTables verwendet. Hierzu werden die ersten fünf Properties jedes Items in tabellarische Form gebracht, um auch verschachtelte Objekte übersichtlich anzeigen zu können. Das Tool ermöglicht es, die geladenen Daten zu sortieren und zu durchsuchen.

Um die Formulare zum Eintragen und Bearbeiten der Daten aus JSON Schema zu generieren, wird das als Open Source Projekt erhältliche Angular-Modul *Angular Schema Form* eingesetzt. Hierüber können die eingegebenen Daten clientseitig entsprechend dem Schema validiert werden. Zudem kann das Modul ebenfalls um weitere Arten von Eingabefeldern erweitert werden, beispielsweise einer Datumsauswahl oder einem Feld für Dateiuploads.

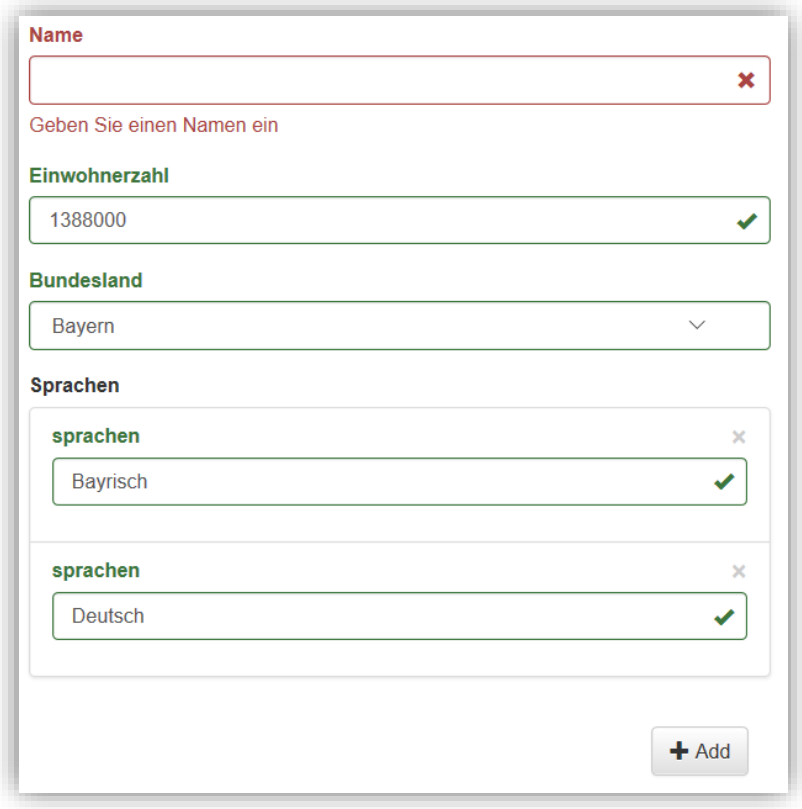

Abbildung 18: Formular entsprechend dem JSON Schema aus Abbildung 16

Schema Form Builder

Im Mai 2015 wurde im GitHub-Repository von Angular Schema Forms erstmals über einen visuellen Editor für JSON Schema diskutiert, um damit Formulare entwerfen zu können. Trotz geäußertem Interesse aus der Community dauerte es bis März 2016, bis ein erster Proof-of-concept von Raphael Owino entstand [75]. Im Rahmen dieser Arbeit wurde diese Webapplikation als Angular-Directive umgeschrieben, um diese einfacher innerhalb anderen Anwendungen verwenden zu können [76]. Parallel dazu wurde die Weiterentwicklung eines Framework-unabhängigen, visuellen JSON-Schema-Generators konzeptionell vorangetrieben.

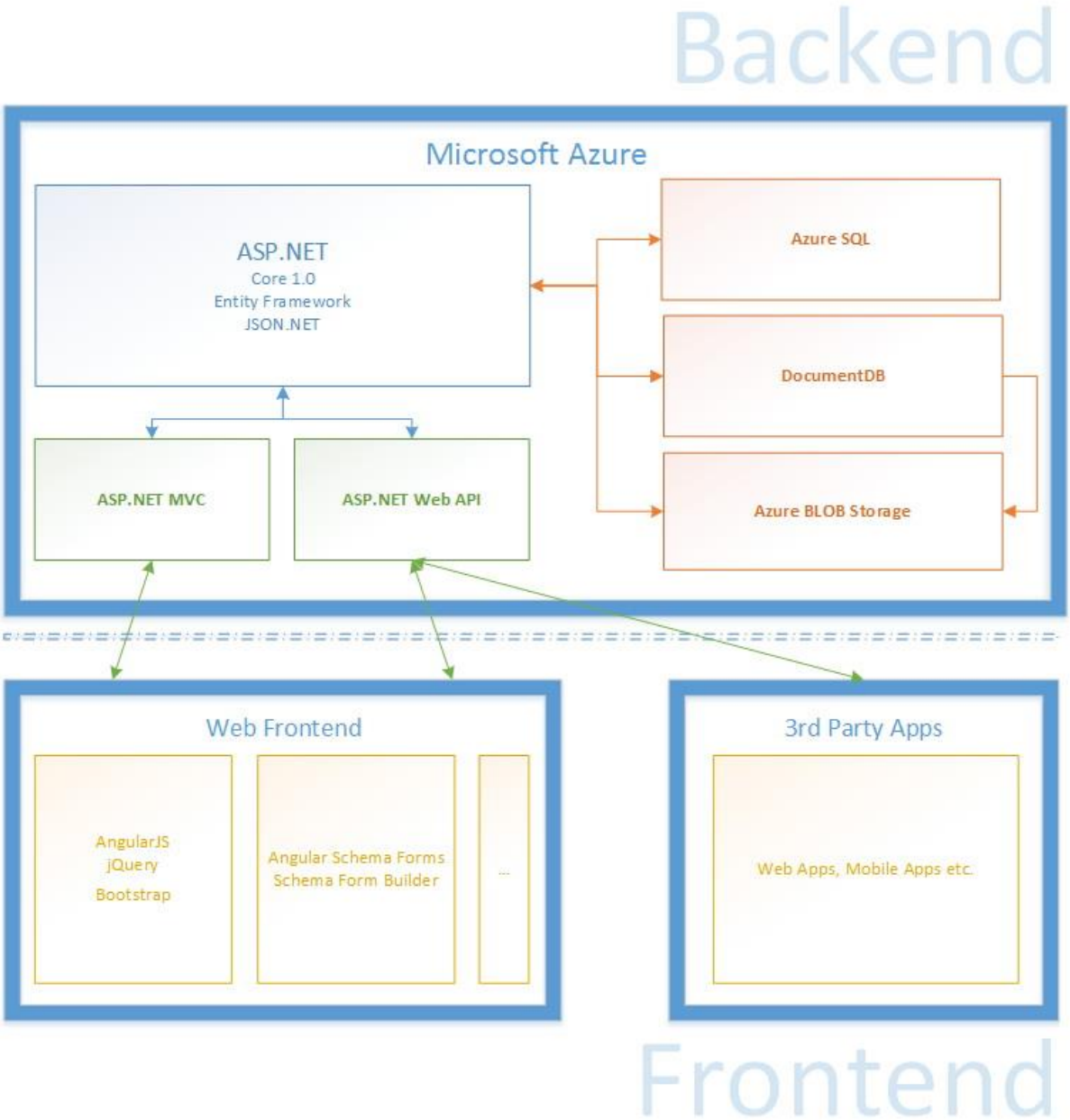

Abbildung 19: Architektur des Prototyps (Produkte)

5.2 Prototyp

Die Kernfunktionalitäten des Prototyps lassen sich in zwei Hauptbereiche unterteilen: das Anlegen von Datensätzen mit der entsprechend gewünschten Datenstruktur sowie Berechtigungen und das anschließende Verwalten der Daten (einfügen, bearbeiten, löschen sowie Änderungen durch andere Personen moderieren). Zusätzlich sind zahlreiche Funktionen enthalten, die als Basis hierfür notwendig sind (Erstellen von Benutzerkonten etc.), auf die an dieser Stelle allerdings nicht näher eingegangen wird.

Erstellung eines Datensatzes

Jeder registrierte und angemeldete Benutzer kann einen Datensatz anlegen. Dieser besteht im Kern aus einem Titel, einer kurzen Beschreibung bezüglich welche Daten gesammelt werden sollen und anderer Hinweise, sowie der Information, ob der Datensatz direkt für jedermann offen zugänglich gemacht werden soll. Außerdem muss definiert werden, wie die Struktur der Daten in diesem Datensatz aussehen muss.

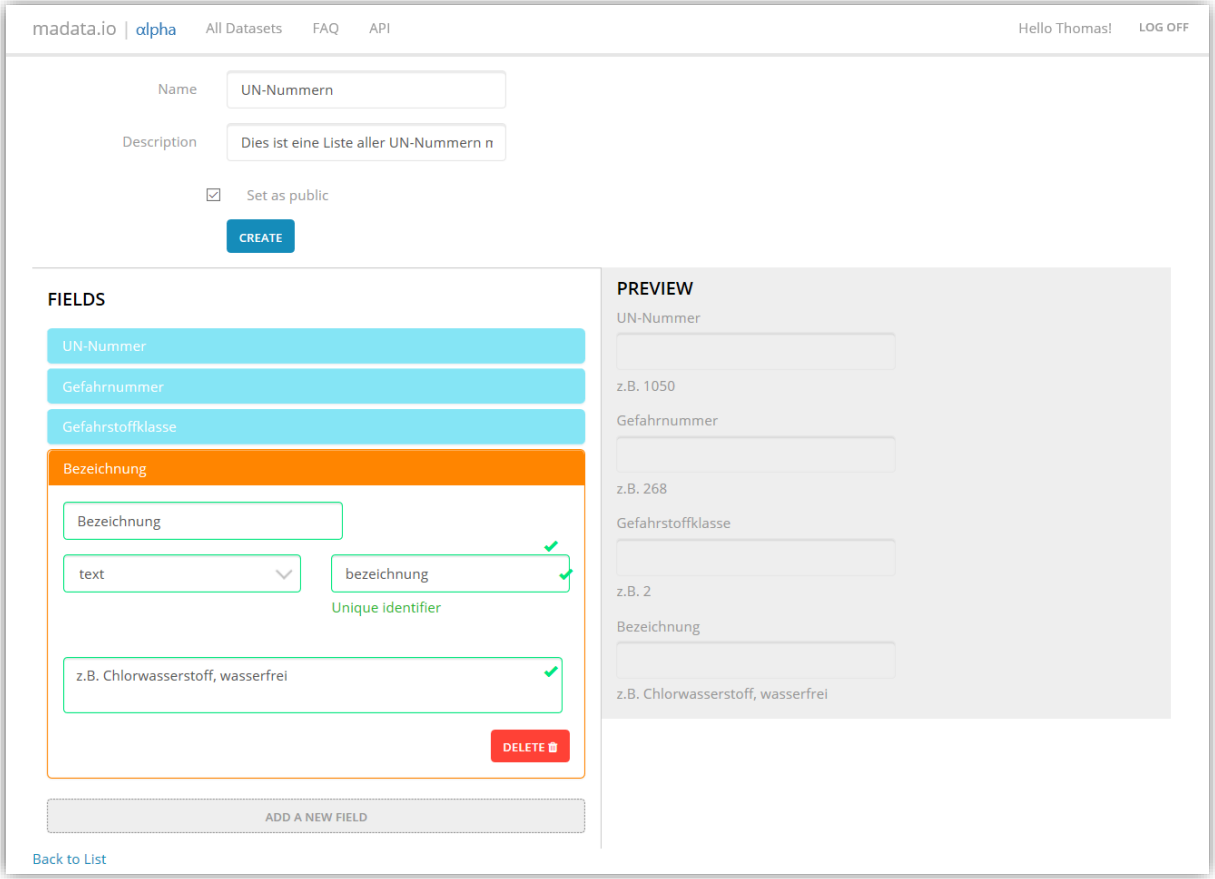

Abbildung 20: Erstellen eines Datensatzes mit Form Builder

Dies wird über einen visuellen Formulareditor realisiert, über den Felder mit entsprechenden Bezeichnungen, Beschreibungen und Datentypen (*text, number, email, boolean, date*) hinzugefügt werden. Diese Formulardefinition wird für den Nutzer unsichtbar als JSON Schema codiert und gespeichert. Abbildung 20: Erstellen eines Datensatzes mit Form Builder zeigt diese Funktionalitäten in einem Screenshot des Prototyps.

Nachdem der Datensatz erstmalig erstellt wurde, können zusätzliche Nutzer als *Contributer* hinzugefügt werden, die somit die Berechtigung zum Ändern und Moderieren von Daten erhalten. Außerdem kann die erstellte Datenstruktur und deren Validierungsregeln von Nutzern mit entsprechenden Kenntnissen direkt durch Bearbeitung des JSON Schemas angepasst werden. Dies kann notwendig sein, da der Formulareditor im aktuellen Stand des Prototyps noch nicht alle Eigenschaften von JSON Schema unterstützt. Diese Funktion ist allerdings nicht sofort Sichtbar und mit einem entsprechenden Warnhinweis versehen, dass ein invalides Schema den Datensatz bzw. das Formular und die Datenvalidierung unbrauchbar machen kann.

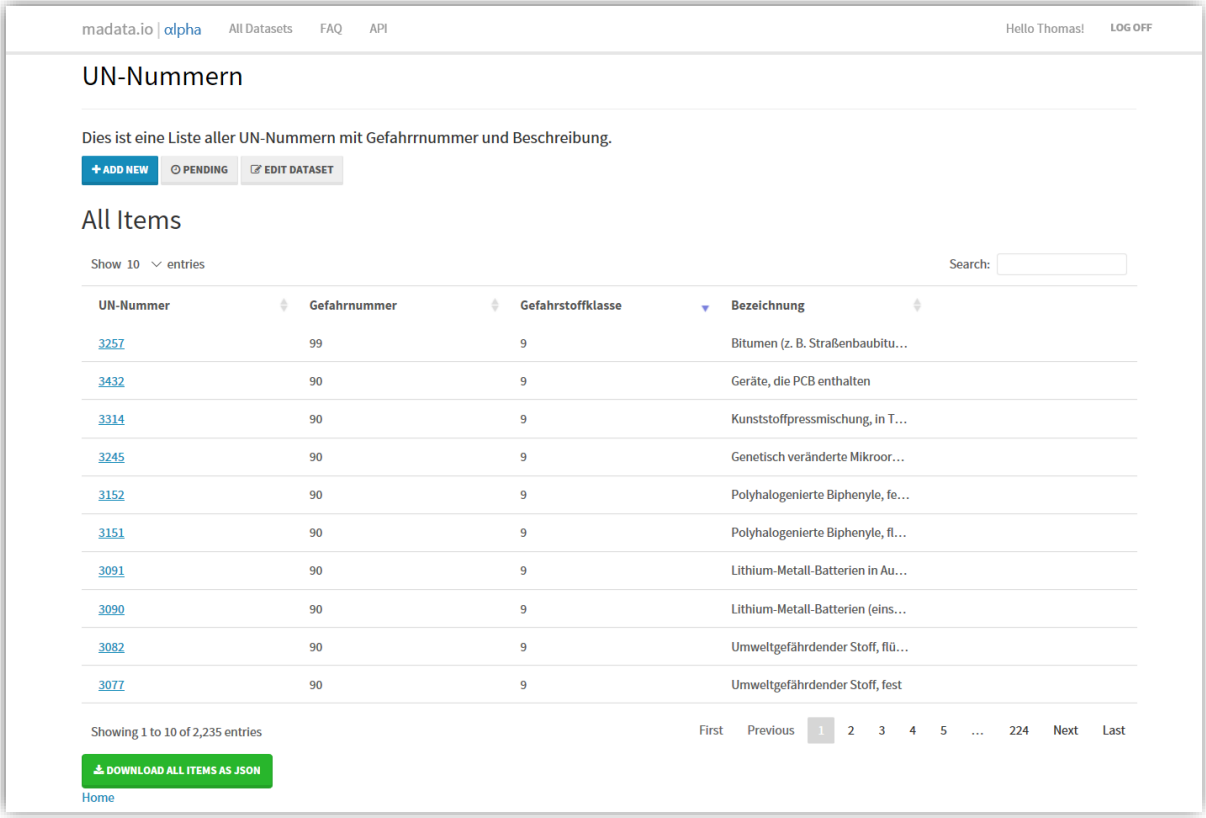

Abbildung 21: Auflistung aller im Datensatz enthaltenen Einträge

Verwaltung der Daten

Anhand der für einen Datensatz definierten Datenstruktur können nun Einträge über das entsprechende Formular (Abbildung 23: Formular zum Hinzufügen und Bearbeiten von Daten) oder auch über die API hinzugefügt oder editiert werden. Diese sind dann in der Übersicht (Abbildung 21: Auflistung aller im Datensatz enthaltenen Einträge) sichtbar und können von dort aus aufgerufen oder gelöscht werden. Werden Daten von Benutzern hinzugefügt oder editiert, die nicht den *Contributor*-Status besitzen, werden diese Änderungen

unter dem Menüpunkt *Pending* aufgeführt und können dort mit den entsprechenden Benutzerrechten moderiert werden (Abbildung 22: Moderationsansicht für ausstehende

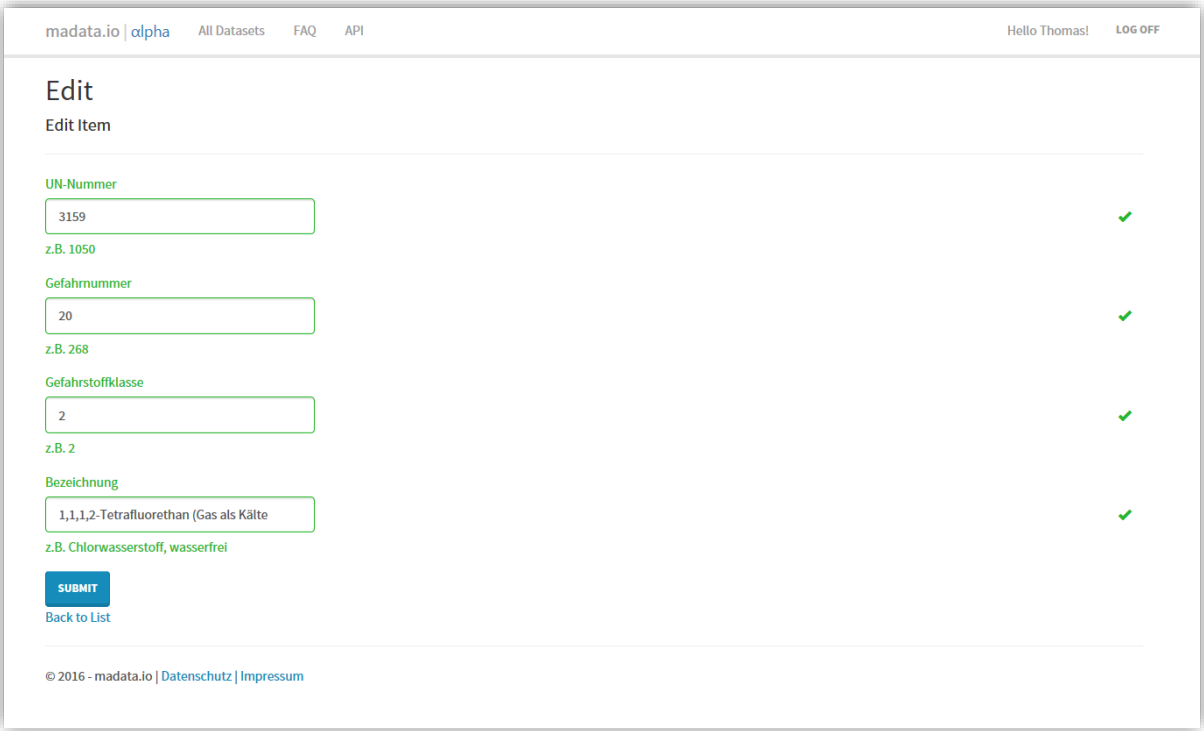

Abbildung 23: Formular zum Hinzufügen und Bearbeiten von Daten

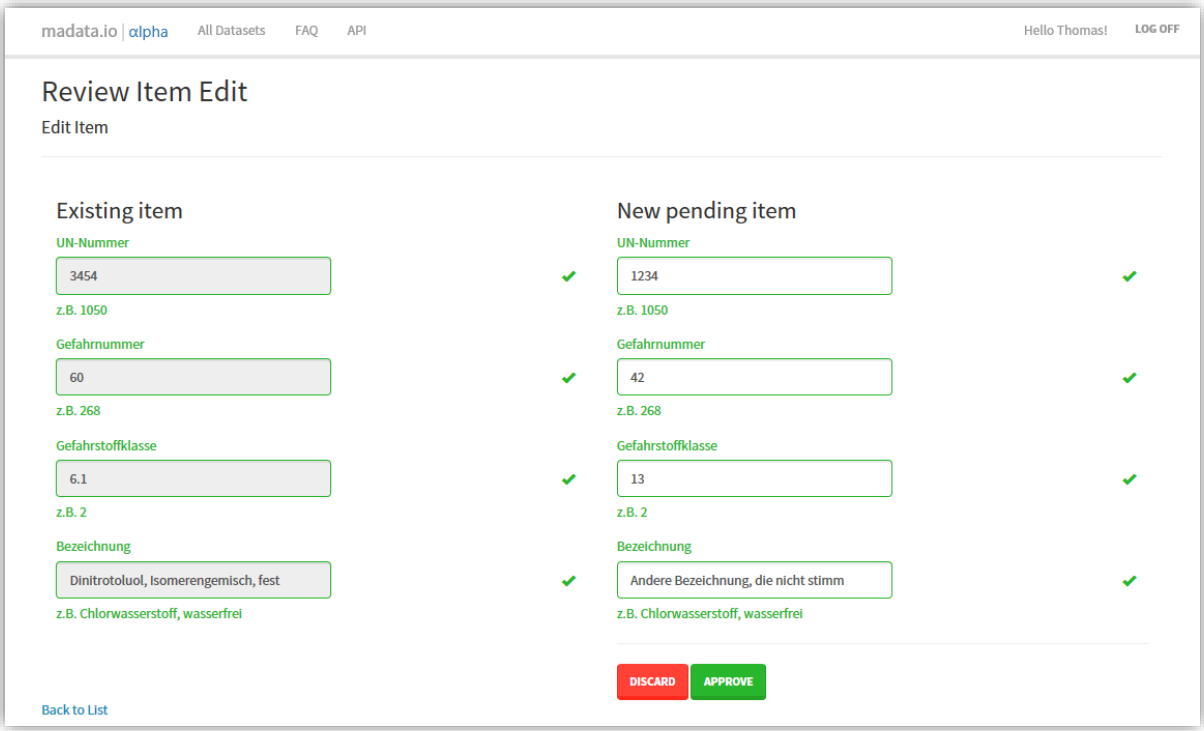

Abbildung 22: Moderationsansicht für ausstehende Änderungen

Änderungen).

Um alle Einträge eines Datensatzes zu extrahieren, kann entweder die Programmierschnittstelle des Tools oder der Button in der Übersicht zum Download aller Daten als JSON-Datei verwendet werden.

swagger http://alpha.madata.io/swagger/v1/swagger.json Explore

API V1

DatasetApi Show/Hide | List Operations | Expand Operations

GET /api
POST /api
GET /api/my
DELETE /api/{id}
GET /api/{id}
PUT /api/{id}

ItemApi Show/Hide | List Operations | Expand Operations

GET /api/{datasetid}/items

Parameters

Parameter	Value	Description	Parameter Type	Data Type
datasetid	(required)		path	integer
pagesize			query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Success		

Try it out!

POST /api/{datasetid}/items
DELETE /api/{datasetid}/items/{itemid}
PUT /api/{datasetid}/items/{itemid}

[BASE URL: / , API VERSION: v1]

VALID {...}

Abbildung 24: Dokumentation der Schnittstelle (API) als swagger Markup

6. Diskussion

In diesem Kapitel wird der Prototyp hinsichtlich des eingangs genannten Ziels der Schaffung einer möglichst einfach zu bedienenden Datenplattform zum gemeinschaftlichen Sammeln, Verwalten und Veröffentlichen von Daten untersucht. Hierfür wurde ein Fragebogen innerhalb des entwickelten Tools in Form eines Datensatzes angelegt. Dieser ist in die drei Hauptpunkte *Dataset Creation*, *Data Administration* und *Data Access* unterteilt, zu denen jeweils folgende Fragen (auf Englisch) gestellt wurden:

- War der Sinn der Funktion ersichtlich? *(Antwortoptionen: ja, nicht sofort, nein)*
- Wie schwierig war es, die Funktion zu nutzen? *(Skala von 1 / schwer bis 10 / einfach)*
- Gab es technische oder konzeptionelle Probleme, und wenn ja, welche? *(Freitext)*
- Würden Sie die Funktion bzw. die Vorgehensweise ändern *(ja, nein)*
- Falls ja, welche Änderungen würden Sie vorschlagen? *(Freitext)*

Zudem wurden zu jedem Fragebogen freiwillige Angaben zu Alter, Geschlecht und Erfahrungen mit Datenbanksystemen sowie Open Data gesammelt. Anhand dieser Informationen ist es möglich einzuschätzen, ob bestimmte Probleme speziell unerfahrene Nutzer betreffen oder ob auch technisch versierte Personen hiermit Schwierigkeiten haben.

Da eine quantitative Befragung im Rahmen dieser Arbeit nicht möglich war, wurden 25 Testpersonen händisch ausgewählt, von denen ca. die Hälfte keine oder kaum Datenbankkenntnisse besitzen. Diese haben den Prototyp mit folgender Aufgabenstellung getestet und anschließend den genannten Fragebogen ausgefüllt:

- Legen Sie einen neuen Datensatz mit Beschreibung und Datenstruktur an.
- Fügen Sie Daten in den neu angelegten Datensatz ein.
- Bearbeiten und löschen Sie einzelne Einträge.
- Moderieren Sie Vorschläge für neue und bearbeitete Einträge.
- Laden Sie die vorhandenen Einträge eines Datensatzes herunter.

Die Auswertung des Feedbacks ist zunächst ebenfalls in dies drei Teilbereiche *Dataset Creation*, *Data Administration* und *Data Access* unterteilt, zu denen die entsprechend angegeben Werte und Anmerkungen aufgearbeitet werden. Zu jedem dieser Bereiche werden zudem die wichtigsten Verbesserungsvorschläge aufgelistet, die vor allem die Bedienungsfreundlichkeit verbessern könnten. Eine vollständige Liste aller

Verbesserungsvorschläge findet sich im Anhang. Die gesammelten Rohdaten liegen in digitaler Form den Anlagen bei.

Dataset Creation

Die Erstellung eines Datensatzes ist sicherlich die wichtigste Funktion des Prototyps und sollte dementsprechend einfach zu handhaben sein. Von allen Testpersonen verstanden lediglich eine gar nicht und weitere sechs nicht sofort, wozu diese Funktion dient. Über 75% der Nutzer hatten hier hingegen keine größeren Schwierigkeiten. Dies spiegelt sich auch in den Bewertungen wieder, wie einfach das Anlegen eines Datensatzes war: im Schnitt 7,2 von 10 möglichen Punkten. Entsprechend gab es auch keine Empfehlungen, die Funktionsweise grundlegend zu ändern, sondern in erster Linie Verbesserungsvorschläge im Bereich der Bedienungsfreundlichkeit:

- Für neue Benutzer wäre es hilfreich, eine detaillierte Schritt-für-Schritt Anleitung anzubieten
- Nachträgliche Änderungen am Datenschema sollten möglich sein. Hierbei muss allerdings eine Lösung gefunden werden, wie mit bestehenden Daten umgegangen wird, die möglicherweise einem geänderten Schema nicht mehr entsprechen und somit auch nicht mehr bearbeitet werden können, da kein passendes Formular hierfür erzeugt wird.
- Dem Benutzer könnte beim ersten Erstellen ein simples Schema als Grundlage bzw. Beispiel vorgegeben werden, anhand dessen die Funktionsweise erlernt werden kann.
- Es sollte nicht möglich sein, Datensätze mit leerem Schema / ohne Felder erstellen zu können.

Data Administration

Die Funktionalitäten zur Datenverwaltung innerhalb des Prototyps wurden von vier Testpersonen nicht auf Anhieb verstanden, von niemandem wurden diese allerdings als völlig unverständlich bezeichnet. Auf der Skala von 1 bis 10 wurde die Simplizität mit 8,24 im Schnitt bewertet. Hauptkritikpunkt war, dass das Konzept der vorgeschlagenen Änderungen (*Pending Items*) nicht näher erklärt wird. Auch hierfür würde sich ein Tutorial anbieten. Die weiteren Änderungsvorschläge beziehen sich vornehmlich auf Usability-Probleme und noch fehlende Funktionen:

- Es sollte nicht möglich sein, leere Einträge abschicken zu können.
- In der Übersicht eines Datensatzes sollte einen Hinweis auf neue, zur Kontrolle ausstehende Einträge angezeigt werden. Zudem wäre eine Benachrichtigung des Datensatz-Besitzers bei neuen Vorschlägen (z.B. via E-Mail) wünschenswert.
- Es fehlt eine Möglichkeit, über Datensätze zu diskutieren und so z.B. auf Mängel oder fehlende Einträge hinweisen zu können.
- Für jeden Datensatz sollte es ein Protokoll aller Aktivitäten geben, um transparent die Änderungen nachvollziehen zu können.

Data Access

Im dritten Teilbereich des Fragebogens wurde abgefragt, wie einfach sich der Zugang zu den Daten gestaltete. Hier hatte ein Tester größere und eine weitere Person kurzzeitige Verständnisschwierigkeiten. Die Funktionen zur Abfrage der Daten wurden im Schnitt mit 8.56 von 10 bewertet und folgende Verbesserungsvorschläge gemacht:

- Einzelne Einträge sollten in der tabellarischen Übersicht vollständig angezeigt werden können, z.B. durch ein Tooltip oder Overlay.
- Für jeden Datensatz und Eintrag sollte der entsprechende API-Endpunkt direkt im Browser mit angezeigt werden.

Allgemeines & Zusammenfassung

Abseits des spezifischen Feedbacks für einzelne Funktionen sind die Rückmeldungen zum Prototyp durchweg positiv. Acht Testpersonen waren noch unentschieden, ob das Tool für sie einen Mehrwert bietet, alle anderen könnten sich die Nutzung in einer stabilen Version vorstellen. Hierfür wäre es auf jeden Fall sinnvoll, einen ausgiebigen Usability Test mit einer höheren Zahl an Testpersonen durchzuführen, um die Benutzung noch einfacher und intuitiver zu gestalten. Um die Einstiegshürde weiter zu senken ist es zudem notwendig, die Anwendung den Nutzern in ihrer jeweiligen Muttersprache zu präsentieren.

Mit Blick auf die oben betrachteten anderen Werkzeuge bringt diese Minimalvariante des vorgestellten Konzepts einige Vorteile mit sich. Durch das Ziel, die Plattform auch Personen nutzbar zu machen, die keine Softwareentwickler sind, ist es beispielsweise im Vergleich zu Github nicht notwendig, sich mit komplexen Themengebieten wie der Versionsverwaltung mit *Git* zu beschäftigen. Im Gegensatz zu ckan ermöglicht das Tool die Pflege der Daten und ist

dabei für beliebige, in sich abgeschlossene Datensätze verwendbar, während WikiData einen riesigen, zusammenhängenden Graphen bildet dessen Struktur nicht eine einzelne Person nach ihren eigenen Anforderungen anpassen kann.

Auch die Anforderungen an Open Data entsprechend Kapitel 3.2 werden alle erfüllt, soweit diese die Bereitstellung der Daten betreffen:

- Datensätze werden unter einer Open Data-konformen Lizenz veröffentlicht.
- Daten sind sowohl für Menschen via Browser als auch maschinell über die API lesbar.
- Daten sind für jedermann ohne Einschränkung und ohne Anmeldezwang zugänglich.
- Alle Daten werden im offenen, nicht-proprietären JSON-Format angeboten.
- Jeder Eintrag besitzt eine eindeutige ID und kann somit direkt verlinkt werden.

Das hier vorgestellte Konzept hat wie gezeigt das Potenzial, einen Mehrwehrt im wachsenden Open-Data-Ökosystem zu liefern. Gerade wo die Werkzeuge mangels Alternativen für Open Data-Zwecke umfunktioniert werden (siehe GitHub) könnte ein hierauf aufbauendes, ausgefeiltes Produkt zum kollaborativen Arbeiten an offenen Datensätzen kann hier bislang bestehende Lücke füllen.

7. Ausblick

Das Konzept von Open Data wird wie in der Einleitung und im Laufe der Arbeit aufgezeigt in jedem Fall an Bedeutung gewinnen, auch weil dessen Prinzipien bereits in vielen etablierten und aufkommenden *Open*-Bewegungen wie Open Government oder Open Innovation stecken.

Es ist gut vorstellbar, dass sich Open Data aus wirtschaftlicher Sicht ähnlich wie Open Source entwickeln könnte. Zu Beginn steht zunächst das Engagement von Aktivisten der Zurückhaltung von Firmen gegenüber, die durch die Veröffentlichung ihrer Daten oder Software einen Wertverlust dieser befürchten. Mittlerweile hat man im Bereich der Softwareentwicklung erkannt, dass Open Source sich auch aus wirtschaftlicher Sicht lohnen kann, z.B. durch kostenlose Fehlersuche durch die Community inklusive konkreter Verbesserungsvorschläge am Code. Durch zahlreiche positive Beispiele (WordPress, MongoDB), wie mit Open Source-Software wirtschaftlich erfolgreich gearbeitet werden kann, ist es für Entscheidungsträger in Firmen einfacher geworden, diesen Weg einzuschlagen und zu rechtfertigen.

Ein solches Positivbeispiel fehlt derzeit noch, damit der Open Data-Gedanken auch aus wirtschaftlicher Sicht volle Anerkennung findet. Vorstellbar ist dies allerdings durchaus. Gerade hinsichtlich der enormen Datenmengen die heutzutage produziert werden kann es durchaus rentabel sein, diese der Öffentlichkeit zugänglich zu machen und so von Analysen, Korrekturen und weiteren darauf basierenden Werken zu profitieren. Bis dahin werden Open Data-Projekte wohl eher eine von Regierungen und Privatpersonen initiierte Sache bleiben.

Das in dieser Arbeit vorgeschlagene Konzept einer Plattform zur Kollaboration an Daten kann allerdings in all den genannten Bereichen nützlich sein, nämlich immer dann, wenn die Datensätze nicht nur veröffentlicht und konsumiert werden sollen, sondern wenn diese Daten mithilfe der Öffentlichkeit vervollständigt, korrigiert oder ein ganz neuer Datensatz aufgebaut werden soll. Gerade wenn es dabei um Themenbereiche geht, in denen Menschen mit unterschiedlichen technischen Fähigkeiten zusammenarbeiten müssen, ist die Anzahl der in Frage kommenden Werkzeuge wie in Kapitel 3.6 gezeigt sehr überschaubar. Wie die Auswertung (siehe Abschnitt 6) gezeigt hat, gibt es besonders im Bereich der Usability noch viele Möglichkeiten zur Verbesserung, die Grundidee wurde jedoch von allen Nutzern verstanden und der Prototyp auch für einen Großteil der Testpersonen ohne größere Hindernisse nutzbar.

Nicht nur für Menschen mit nicht-technischem Hintergrund, auch für beispielsweise Softwareentwickler können diese Art von Werkzeugen viel Initialaufwand für ein neues Projekt ersparen, zumal mit nur wenig Aufwand Daten kollaborativ gesammelt und bearbeitet werden können, ohne dass für die eigentliche Anwendung eine Infrastruktur (Server, Datenbanken) oder Code bereitstehen müssen.

8. Zusammenfassung

In dieser Arbeit wurde das Thema Open Data und hierfür verfügbare Werkzeuge betrachtet. Zunächst folgte ein Überblick darüber, was Daten sind und wie Open Data definiert werden kann. Anschließend wurde festgestellt, dass es bislang schwierig ist, gemeinschaftlich an offenen Datensätzen zu arbeiten. Bestehende Open Data-Tools entweder nur auf die Veröffentlichung von Daten spezialisiert oder die Anwendungen für technisch versierte Nutzer konzipiert und stellen somit eine hohe Einstiegshürde für normale Benutzer dar.

Es wurde ein Konzept für eine Anwendung vorgeschlagen, die das Sammeln und Bearbeiten von strukturierten Daten ohne Datenbank oder sonstigen technischen Kenntnissen erleichtern soll. Im Kern steht die Idee, über einen visuellen Editor Formulare bauen zu können, welche gleichzeitig das Eintragen und Editieren von Daten erleichtern und zudem eine feste Struktur der Daten vorgeben. Durch ein entsprechendes Berechtigungskonzept können diese Datensätze von allen Nutzern eingesehen und Änderungsvorschläge sowie neue Einträge übermittelt werden, die vom Besitzer des Datensatzes gesichtet und akzeptiert oder gelöscht werden können. Um den gängigen Open Data-Richtlinien zu entsprechen sind die Daten sowohl für Menschen über den Browser als auch für Maschinen via HTTP-Schnittstelle (API) zugänglich.

Der Prototyp wurde in Form eines Fragebogens bezüglich der Erfahrungen der Benutzer evaluiert. Hieraus ergab sich, dass der verfolgte Ansatz vielversprechend ist und hinsichtlich der existierenden Lösungen einen Mehrwert bietet. Verbesserungsbedarf gibt es vor allem bezüglich der Bedienbarkeit der Funktionen, um die Benutzung des Werkzeugs noch einfacher und intuitiver zu gestalten.

Glossar

Abgeleitete Daten 13
ACID 49
Application Programming Interface (API) 56
ASP.NET Core 55
Attribuierte Daten 14
BASE 49
Big Data 23
CKAN 42
Column Family Stores 50
CSV 15
Daten 10
Datenformate 14
Document Databases 51
DocumentDB 56
GitHub 44
Graph Databases 51
Indexierte Daten 14
JSON 17
JSON Schema 51
JSON-LD 21
Key-Value Stores 50
Linked Data 18
Metadaten 14
NoSQL 49
Open Access 28
Open Data 29
Open Education 29
Open Government 28
Open Source 27
Primäre Daten 13
Qualitative Daten 12
Quantitative Daten 11
RDF 19
RDF+XML 19
RDFa 20
REST 53
Sekundäre Daten 13
Small Data 22
SPARQL Query Language 21
Strukturierte Daten 12
Teilweise strukturierte Daten 12
Tertiäre Daten 14
Transienten Daten 13
Unstrukturierte Daten 12
Wikidata 45
XML 16

Anhang

Verbesserungsvorschläge für Prototyp

Dataset Creation

- Für neue Benutzer wäre es hilfreich, eine detaillierte Schritt-für-Schritt Anleitung anzubieten
- Nachträgliche Änderungen am Datenschema sollten möglich sein. Hierbei muss allerdings eine Lösung gefunden werden, wie mit bestehenden Daten umgegangen wird, die möglicherweise einem geänderten Schema nicht mehr entsprechen und somit auch nicht mehr bearbeitet werden können, da kein passendes Formular hierfür erzeugt wird.
- Der Formulargenerator könnte vereinfacht werden, indem die Key-Felder (Unique Identifier) automatisch z.B. aus dem Feldnamen generiert werden.
- Dem Benutzer könnte beim ersten Erstellen ein simples Schema als Grundlage bzw. Beispiel vorgegeben werden, anhand dessen die Funktionsweise erlernt werden kann.
- Es sollte nicht möglich sein, Datensätze mit leerem Schema / ohne Felder erstellen zu können.
- Es sollte eine entsprechende Open-Data-kompatible Lizenz beim Erstellen des Datensatzes ausgewählt werden können und nicht pauschal alles unter der Open Data Commons Public Domain Dedication and Licence (ODC PDDL) Lizenz veröffentlicht werden.
- Die Unterstützung zusätzlicher Feldertypen (Uhrzeit, Farben, RegEx) und ein Dateiupload für Bilder und Dokumente sind wünschenswert.

Dataset Administration

- Im Datepicker für die Datumsauswahl sollte es möglich sein, Jahreszahlen direkt eingeben zu können anstatt nur Monatsweise vorwärts oder rückwärts springen zu können.
- Es sollte nicht möglich sein, leere Einträge abschicken zu können.
- Eigene Änderungsvorschläge an fremden Datensätzen sollte man nachträglich bearbeiten können.

- In der Übersicht eines Datensatzes sollte einen Hinweis auf neue, zur Kontrolle ausstehende Einträge angezeigt werden. Zudem wäre eine Benachrichtigung des Datensatz-Besitzers bei neuen Vorschlägen (z.B. via E-Mail) wünschenswert.
- Es fehlt eine Möglichkeit, über Datensätze zu diskutieren und so z.B. auf Mängel oder fehlende Einträge hinweisen zu können.
- Einzelne Einträge sollten als mangelhaft markiert werden können, um eine Korrektur anzuregen. Der Besitzer sollte zudem korrekte Einträge oder komplette Datensätze für Änderungsvorschläge sperren können.
- Die interne Versionierung einzelner Einträge sollte auch im Browser einsehbar sein, um diese wiederherstellen zu können.
- Für jeden Datensatz sollte es ein Protokoll aller Aktivitäten geben, um transparent die Änderungen nachvollziehen zu können.

Data Access

- Einzelne Einträge sollten in der tabellarischen Übersicht vollständig angezeigt werden können, z.B. durch ein Tooltip oder Overlay.
- Der zugehörige Titel des aktuell geöffneten Datensatzes sollte im Titel des Browserfensters angezeigt werden, um die parallele Verwaltung mehrerer Datensätze zu vereinfachen.
- Für jeden Datensatz und Eintrag sollte der entsprechende API-Endpunkt direkt im Browser mit angezeigt werden.

Literaturverzeichnis

[1] D. B. W. P. H. K. Marcus M. Dapp, „Open Data. The Benefits - Das volkswirtschaftliche Potential für Deutschland," 2016. [Online]. Available: http://www.kas.de/wf/doc/kas_44906-544-1-30.pdf?160418125028. [Zugriff am 25 10 2016].

[2] Duden, „Duden Rechtschreibung," [Online]. Available: http://www.duden.de/rechtschreibung/Daten. [Zugriff am 11 09 2015].

[3] R. Kitchin, The Data Revolution, London: SAGE Publications Ltd, 2014.

[4] D. Rosenberg, in *"Raw data" is an oxymoron*, Cambridge, Massachusetts, The MIT Press, 2013.

[5] L. Floridi, „http://www.philosophyofinformation.net/," 2008. [Online]. Available: http://www.philosophyofinformation.net/wp-content/uploads/sites/67/2014/05/data.pdf. [Zugriff am 11 09 2016].

[6] C. E. G. L. P. Z. T. D. Dirk deRoos, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw-Hill, 2011.

[7] http://uni-graz.at, „Zentrum für Informationsmodellierung in den Geisteswissenschaften," [Online]. Available: http://www-gewi.uni-graz.at/zim/lehre/datenformat.html. [Zugriff am 31 10 2016].

[8] inf-schule.de, „Exkurs - Datenformate," 24 07 2014. [Online]. Available: http://inf-schule.de/information/informationsdarstellungxml/xmlsprachen/exkurs_datenformat. [Zugriff am 31 10 2016].

[9] Y. Shafranovich, „Common Format and MIME Type for Comma-Separated Values (CSV) Files," 10 2005. [Online]. Available: https://tools.ietf.org/html/rfc4180. [Zugriff am 12 09 2016].

[10] J. P. C. M. S.-M. Tim Bray, „Extensible Markup Language (XML) 1.0,“ W3C, 10 02 1998. [Online]. Available: https://www.w3.org/TR/1998/REC-xml-19980210. [Zugriff am 12 09 2016].

[11] T. Bray, „The JavaScript Object Notation (JSON) Data Interchange Format,“ 03 2014. [Online]. Available: https://tools.ietf.org/html/rfc7159 und ECMA-404. [Zugriff am 11 09 2016].

[12] ecma International, „The JSON Data Interchange Format,“ 10 2013. [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf. [Zugriff am 11 09 2016].

[13] „Introducing JSON,“ [Online]. Available: http://json.org/. [Zugriff am 13 09 2016].

[14] T. Berners-Lee, „Linked Data,“ 27 7 2006. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html. [Zugriff am 06 09 2016].

[15] W3C, „Resource Description Framework (RDF),“ 10 02 2004. [Online]. Available: https://www.w3.org/2001/sw/wiki/RDF. [Zugriff am 06 09 2016].

[16] W3C, „RDF 1.1 XML Syntax,“ 25 02 2014. [Online]. Available: http://www.w3.org/TR/rdf-syntax-grammar/. [Zugriff am 06 09 2016].

[17] OpenOrgs, „Open Data Patterns for Organisations,“ 25 03 2015. [Online]. Available: http://openorg.ecs.soton.ac.uk/wiki/Linked_Data_Basics_for_Techies#Semantic_Overflow. [Zugriff am 06 09 2016].

[18] LinkedDataTools, „LinkedDataTools.com,“ [Online]. Available: http://www.linkeddatatools.com/introducing-rdf-part-2. [Zugriff am 06 09 2016].

[19] CreativeCommons, „rdfa.info,“ [Online]. Available: https://rdfa.info/. [Zugriff am 06 09 2016].

[20] W3C, „RDFa 1.1 Primer - Third Edition,“ 17 03 2015. [Online]. Available: http://www.w3.org/TR/rdfa-primer/. [Zugriff am 06 09 2016].

[21] W3C, „RDFa Lite 1.1 - Second Edition,“ 17 03 2015. [Online]. Available: http://www.w3.org/TR/rdfa-lite/. [Zugriff am 07 09 2016].

[22] „JSON-LD,“ [Online]. Available: http://json-ld.org/. [Zugriff am 07 09 2016].

[23] W3C, „SPARQL Query Language for RDF,“ 15 01 2008. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/. [Zugriff am 07 09 2016].

[24] C. B. A. J. R. C. Max Schmachtenberg, „Linking Open Data cloud diagram 2014,“ 2014. [Online]. Available: http://lod-cloud.net/. [Zugriff am 07 09 2016].

[25] jworkorg, „"Small data". Never heard this term?,“ 01 02 2014. [Online]. Available: http://jwork.org/main/node/18. [Zugriff am 29 09 2016].

[26] A. Bonde , „Small Data Group,“ 18 10 2013. [Online]. Available: https://smalldatagroup.com/2013/10/18/defining-small-data/. [Zugriff am 29 09 2016].

[27] J. Thomas Barnett, „The Dawn of the Zettabyte Era,“ 23 06 2011. [Online]. Available: http://blogs.cisco.com/news/the-dawn-of-the-zettabyte-era-infographic. [Zugriff am 28 09 2016].

[28] K. W. Pamela Vagata, „Scaling the Facebook data warehouse to 300 PB,“ 10 04 2014. [Online]. Available: https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/. [Zugriff am 28 09 2016].

[29] K. Smith, „Brandwatch: 44 Twitter Statistics for 2016,“ 17 05 2015. [Online]. Available: https://www.brandwatch.com/2016/05/44-twitter-stats-2016/. [Zugriff am 28 09 2016].

[30] CERN, „Processing: What to record?,“ [Online]. Available: https://home.cern/about/computing/processing-what-record. [Zugriff am 28 09 2016].

[31] O. S. Initiative, „opensource.org,“ 09 2012. [Online]. Available: https://opensource.org/history. [Zugriff am 04 10 2016].

[32] O. S. Initiative, „opensource.org,“ 22 03 2007. [Online]. Available: https://opensource.org/docs/osd. [Zugriff am 04 10 2016].

[33] P. D. J. v. Lucke, „Öffnung von Staat und Verwaltung,“ 09 05 2010. [Online]. Available: https://www.zu.de/institute/togi/assets/pdf/JvL-100509-Open_Government-V2.pdf. [Zugriff am 04 10 2016].

[34] J. H. U. K. J. v. B. O. M. H.-D. Z. Jörn von Lucke, „eSociety Bodensee 2020,“ 15 09 2012. [Online]. Available: https://esocietybodensee2020.wordpress.com/publikationen/offene-gesellschaftliche-innovation-die-seealemannische-definition/. [Zugriff am 04 10 2016].

[35] open-acces.net, „Was bedeutet Open Access?,“ [Online]. Available: https://open-access.net/informationen-zu-open-access/was-bedeutet-open-access/. [Zugriff am 05 10 2016].

[36] open-access.org, „Gründe und Vorbehalte - Argumente für Open Access,“ [Online]. Available: https://open-access.net/informationen-zu-open-access/gruende-und-vorbehalte/. [Zugriff am 05 10 2016].

[37] „Open Defintion,“ Open Knowledge Foundation, [Online]. Available: http://opendefinition.org/od/2.1/en/. [Zugriff am 05 09 2016].

[38] P. Carl Malamud, „Open Government Working Group,“ 22 10 2007. [Online]. Available: https://public.resource.org/open_government_meeting.html. [Zugriff am 05 09 2016].

[39] „Open Government Data Principles,“ 08 12 2007. [Online]. Available: https://public.resource.org/8_principles.html. [Zugriff am 05 09 2016].

[40] OpenGovData, „The Annotated 8 Principles of Open Government Data,“ [Online]. Available: https://opengovdata.org/. [Zugriff am 05 09 2016].

[41] S. Foundation, „Open Data Policy Guidelines,“ [Online]. Available: http://sunlightfoundation.com/opendataguidelines/. [Zugriff am 11 09 2016].

[42] T. Berners-Lee, „5StarData.info,“ 22 01 2012. [Online]. Available: http://5stardata.info/de/. [Zugriff am 08 09 2016].

[43] O. D. Insitute, „Open Data Certificate,“ [Online]. Available: https://certificates.theodi.org. [Zugriff am 11 09 2016].

[44] Open Definition, „Non-Conformant Licenses,“ [Online]. Available: http://opendefinition.org/licenses/nonconformant/. [Zugriff am 05 10 2016].

[45] OpenDefinition, „Open Definition: Conformant Licenses,“ [Online]. Available: http://opendefinition.org/licenses/. [Zugriff am 06 09 2016].

[46] J. Fischer, „niederrheinzeitung.de,“ [Online]. Available: http://www.niederrheinzeitung.de/aktuell/wirtschaft-finanzen/9281-opendata-die-daten-gehoeren-den-menschen. [Zugriff am 23 10 2016].

[47] „Gesetz über Urheberrecht und verwandte Schutzrechte (Urheberrechtsgesetz), § 1 Allgemeines,“ [Online]. Available: https://www.gesetze-im-internet.de/urhg/__1.html. [Zugriff am 13 09 2016].

[48] G. Gordon, „Towards an overarching narrative on open government,“ 12 02 2013. [Online]. Available: http://www.opengovernment.org.uk/2013/02/12/towards-an-overarching-narrative-on-open-government/. [Zugriff am 13 09 2016].

[49] M. A. P. B. K. Harry Verwayen, „The Problem of the Yellow Milkmaid A Business Model Perspective on Open Metadata,“ 11 2011. [Online]. Available: http://pro.europeana.eu/files/Europeana_Professional/Publications/Whitepaper_2-The_Yellow_Milkmaid.pdf. [Zugriff am 14 09 2016].

[50] K. Janssen, „Open Government Data: Right to Information 2.0 or its Rollback Version?,“ 1 09 2012. [Online]. Available: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2152566. [Zugriff am 14 09 2016].

[51] M. Grüebler, „Open Data der Stadt Zürich. Was bringt es?,“ 27 11 2014. [Online]. Available: http://de.slideshare.net/Opendatazurich/open-data-der-stadt-zrich-was-bringt-es. [Zugriff am 13 09 2016].

[52] S. F. E. v. S. Wendy Carrara, „Open Data Maturity in Europe 2015,“ 2015. [Online]. Available: Open Data Maturity in Europe 2015. [Zugriff am 14 09 2016].

[53] B. Wellington, „I Quant NY,“ 11 05 2016. [Online]. Available: http://iquantny.tumblr.com/post/144197004989/the-nypd-was-systematically-ticketing-legallyBen Wellington. [Zugriff am 14 09 2016].

[54] E. Panek, „Open Knowledge Foundation Deutschland,“ 30 11 2012. [Online]. Available: https://okfn.de/blog/2012/11/7-reasons-against-open-data/. [Zugriff am 24 10 2016].

[55] K. S. Frank Bräutigam, „Tagesschau.de,“ 30 04 2015. [Online]. Available: https://www.tagesschau.de/inland/tagesschau-app-bgh-faq-101.html. [Zugriff am 14 09 2016].

[56] M. F. S. T. a. M. I. Sébastien Martin, „Risk Analysis to Overcome Barriers to Open Data,“ [Online]. Available: www.ejeg.com/issue/download.html?idArticle=296. [Zugriff am 24 10 2016].

[57] D. M. J. Karen EC Levy, „When open data is a Trojan Horse: The weaponization of transparency in science and governance,“ 03 2016. [Online]. Available: http://bds.sagepub.com/content/3/1/2053951715621568#sec-2. [Zugriff am 14 09 2016].

[58] J. A. Spertus, „The Double-Edged Sword of Open Access to Research Data,“ Circulation: Cardiovascular Quality and Outcomes, 05 2012. [Online]. Available: http://circoutcomes.ahajournals.org/content/5/2/143.full. [Zugriff am 14 09 2016].

[59] A. G.-S. Y. K. e. a. Ron A. M. Fouchier, „Pause on avian flu transmission studies,“ 20 01 2012. [Online]. Available: http://www.nature.com/nature/journal/v481/n7382/full/481443a.html. [Zugriff am 14 09 2016].

[60] V. Pandurangan, „On Taxis and Rainbows ,“ 21 06 2014. [Online]. Available: https://tech.vijayp.ca/of-taxis-and-rainbows-f6bc289679a1#.p2ainigsw. [Zugriff am 14 09 2016].

[61] D. I. Sommer, „Open Data zwischen Datenschutz und Informationsfreiheit,“ 11 11 2011. [Online]. Available:

transparenz.bremen.de/sixcms/media.php/13/OpenData_Datenschutz.pdf. [Zugriff am 24 10 2016].

[62] S. Sobola, „manager magazin - Wer haftet, wenn es schief geht?," 11 10 2004. [Online]. Available: http://www.manager-magazin.de/unternehmen/it/a-322293.html. [Zugriff am 14 09 2016].

[63] „ckan," [Online]. Available: http://ckan.org/. [Zugriff am 15 09 2016].

[64] „Wikidata Help:Items," [Online]. Available: https://www.wikidata.org/wiki/Help:Items. [Zugriff am 16 09 2016].

[65] U. N. E. C. f. Europe, „UNECE.org," [Online]. Available: http://www.unece.org/trans/danger/publi/unrec/rev19/19files_e.html. [Zugriff am 07 09 2016].

[66] P. D. S. Edlich, „nosql-database.org," [Online]. Available: http://nosql-database.org. [Zugriff am 26 09 2016].

[67] D. E. A. Brewer, „Towards Robust Distributed Systems," 19 07 2000. [Online]. Available: https://people.eecs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf. [Zugriff am 26 09 2016].

[68] G. Zerbes, „Datenbanktechnologien – ACID, BASE, CAP und Google Spanner," 25 04 2014. [Online]. Available: http://www.norcom.de/de/fachartikel/datenbanktechnologien-acid-base-cap-und-google-spanner. [Zugriff am 26 09 2016].

[69] M. Vardanyan, „Picking the Right NoSQL Database Tool," 22 05 2011. [Online]. Available: http://www.monitis.com/blog/picking-the-right-nosql-database-tool/. [Zugriff am 26 09 2016].

[70] e. a. Galiegue, „ JSON Schema: core definitions and terminology draft-zyp-json-schema-04," 31 01 2013. [Online]. Available: https://tools.ietf.org/html/draft-zyp-json-schema-04. [Zugriff am 23 09 2016].

[71] „JSON Schema,“ [Online]. Available: http://json-schema.org/documentation.html. [Zugriff am 23 09 2016].

[72] T. Bayer, 11 2002. [Online]. Available: http://www.oio.de/public/xml/rest-webservices.htm. [Zugriff am 08 10 2016].

[73] Wikipedia, „ASP.NET,“ [Online]. Available: https://de.wikipedia.org/wiki/ASP.NET. [Zugriff am 27 09 2016].

[74] Microsoft, „ASP.NET,“ [Online]. Available: http://www.asp.net/. [Zugriff am 27 09 2016].

[75] F. Köln, „Datenbanken Online Lexikon,“ 23 01 2015. [Online]. Available: http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/CAP. [Zugriff am 20 09 2016].

[76] Microsoft, „Consistency levels in DocumentDB,“ 2016 08 24. [Online]. Available: https://azure.microsoft.com/en-us/documentation/articles/documentdb-consistency-levels/. [Zugriff am 20 09 2016].

[77] O. A. Initiative, „The OpenAPI Specification (fka The Swagger Specification),“ 22 08 2016. [Online]. Available: https://github.com/OAI/OpenAPI-Specification. [Zugriff am 20 09 2016].

[78] R. Owino, „Schema Form Builder,“ 08 03 2016. [Online]. Available: https://github.com/ralphowino/schema-form-builder. [Zugriff am 20 09 2016].

[79] T. Pentenrieder, „Merge pull request #6 from ThomasPe/directive,“ 13 06 2016. [Online]. Available: https://github.com/json-schema-form/json-schema-builder/commit/be91456a625ebd60bafcbfcc6bebd277d3af9046. [Zugriff am 20 09 2016].

[80] SunlightFoundation, „Open Data Policy Guidelines,“ [Online]. Available: http://sunlightfoundation.com/opendataguidelines/. [Zugriff am 06 09 2016].

[81] Masur, „Eigenes Werk based on European Agreement concerning the International Carriage of Dangerous Goods by Road; Part 5,“ 01 04 2010. [Online]. Available:

https://commons.wikimedia.org/w/index.php?curid=1596635. [Zugriff am 08 09 2016].

[82] „Wikipedia,“ [Online]. Available: https://de.wikipedia.org/wiki/T%C3%BCbingen. [Zugriff am 16 09 2016].

Anlagen

- Eine digitale Kopie dieser Arbeit
- Der Source Code des Prototyps als Solution für Visual Studio
- Die erstellten Visio-Diagramme
- Rohdaten aus dem Feedbackformular