Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Nutzung von Wissensgraphen zur Vermittlung von Daten zwischen Produzenten und Konsumenten auf Datenplattformen

Ronja Sölch

Studiengang: Wirtschaftsinformatik

Prüfer/in: Prof. Dr. rer. nat. habil. Holger Schwarz

Betreuer/in: Jan Schneider, M.Sc.

Beginn am: 18. August 2024

Beendet am: 18. Februar 2025

Kurzfassung

Die vorliegende Arbeit untersucht, inwieweit der Einsatz eines Wissensgraphen die Vermittlung von Daten zwischen Datenproduzenten und -konsumenten optimieren kann. Im Rahmen dieser Untersuchung wurden zunächst die spezifischen Anforderungen beider Akteursgruppen systematisch durch eine Literaturanalyse erfasst. Auf Basis dieser Analyse wurde ein mehrschichtiger Wissensgraph konzipiert, der es den Datenkonsumenten ermöglicht, sogenannte *Bedarfsknoten* zu erstellen. Diese Bedarfsknoten repräsentieren die spezifischen Anforderungen der Konsumenten, indem sie zum einen Attribute zur präzisen Beschreibung des Bedarfs und zum anderen Relationen zu den relevanten Entitäten im Wissensgraphen definieren. Datenproduzenten erhalten so einen strukturierten Einblick in die geäußerten Bedürfnisse und können darauf basierend die entsprechenden Daten bereitstellen. Die erfolgreiche Implementierung des Konzepts belegt, dass die Integration von Bedarfsknoten die Datenvermittlung effizient fördert. Gleichzeitig wurden jedoch Limitationen identifiziert, da nicht sämtliche der erfassten Anforderungen vollständig berücksichtigt werden konnten. Die Arbeit liefert damit eine Grundlage für weiterführende Forschungen zur Optimierung wissensgraphbasierter Datenvermittlung und zeigt potenzielle Erweiterungsmöglichkeiten auf.

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Hinführung und Problemstellung . . . 1
1.2 Zielsetzung . . . 2

2 Grundlagen **3**
2.1 Datenplattformen . . . 3
2.1.1 Data Lakehouse . . . 4
2.1.2 Datenkatalog . . . 6
2.1.3 Zonenreferenzmodell . . . 8
2.2 Wissensgraph . . . 9

3 Verwandte Arbeiten **13**
3.1 Datenkatalog . . . 13
3.2 Literatur . . . 14

4 Datenkonsumenten und Datenproduzenten **17**
4.1 Aufgaben . . . 17
4.1.1 Data Owner . . . 18
4.1.2 Data Steward . . . 19
4.1.3 Data Engineer . . . 19
4.1.4 Data Scientists . . . 20
4.2 Anforderungen . . . 21

5 Konzeption eines Wissensgraphen **25**
5.1 Entwurf eines Wissensgraphen . . . 25
5.1.1 Aufbau des Wissensgraphen . . . 26
5.1.2 Semantische Ebene . . . 28
5.1.3 Data Lineage . . . 28
5.1.4 Akteure und Rollen . . . 29
5.2 Interaktion mit dem Wissensgraphen . . . 30
5.2.1 Benachrichtigungen bei Änderungen des Wissensgraphen . . . 30
5.2.2 Suche im Wissensgraph . . . 31
5.2.3 Aktualisierung des Wissensgraph . . . 32
5.3 Bedarfsknoten . . . 32
5.3.1 Bedarfsknoten-Ontologie . . . 33
Attribute . . . 33
Relationen . . . 34
Einschränkungen . . . 35
5.3.2 Weitere Arten von Bedarfsknoten . . . 36
Bedarfsknoten auf der Datensatzreferenzebene . . . 36

Bedarfsknoten auf der Metadatenebene 37
Bedarfsknoten auf der Domänenebene 38
5.3.3 Verwendung des Bedarfsknotens 39
5.3.4 Lebenszyklus von Bedarfsknoten 40
5.4 Interaktion von Akteuren mit dem Wissensgraph 40

6 Implementierung 43
6.1 Technologien 43
6.2 Datensatz 44
6.2.1 Domänenebene 44
6.2.2 Datensatzreferenzebene 47
6.2.3 Datenproduktebene 48
6.2.4 Metadatenebene 49
6.3 Benutzeroberfläche 51
6.3.1 Erstellung von Bedarfsknoten 51
6.3.2 Benutzereingabe 54
6.3.3 Änderung des Status des Bedarfsknoten 55
6.3.4 Suche im Wissensgraph 55
6.3.5 Einfluss der Benutzerrollen 57
6.3.6 Attribut von Knoten 58

7 Diskussion 61
7.1 Evaluation 61
7.2 Limitation 64

8 Fazit und Ausblick 67
8.1 Zusammenfassung 67
8.2 Fazit 67
8.3 Ausblick 68

9 Anhang 71

Literaturverzeichnis 73

Abbildungsverzeichnis

2.1 Schematische Architektur eines Data Warehouses und eines Data Lake 3
2.2 Schemantische Architektur eines Data Lakehouse 5
2.3 Zonenreferenzmodell 8
2.4 Beispielhafte Darstellung eines Wissensgraphen 10

5.1 Aufbau des Wissensgraphen 27
5.2 Darstellung der Akteursknoten und ihrer Interaktion mit Entitätsknoten 30
5.3 Darstellung der Verknüpfung eines Akteurs mit einer Entität, an der Interesse besteht 31
5.4 Struktur eines Bedarfsknotens auf der Datenproduktebene 33
5.5 Visualisierung der Struktur eines Bedarfsknotens auf Datenproduktebene 34
5.6 Automatische Erkennung des Datenproduzenten bei Erstellung des Bedarfsknoten 35
5.7 Struktur eines Bedarfsknotens auf der Datensatzreferenzebene im Kontext der Datenqualität 36
5.8 Aufbau eines Bedarfsknoten auf der Datensatzreferenzebene 37
5.9 Aufbau eines Bedarfsknoten auf der Metadatenebene 37
5.10 Struktur eines Bedarfsknotens auf der Metadatenebene zur Metadatenanfrage . . 38
5.11 Struktur eines Bedarfsknoten auf der Domänenebene 38
5.12 Aufbau eines Bedarfsknoten auf der Domänenebene 39
5.13 Ablaufdiagramm eines Data Scientists bei der Nutzung eines Wissensgraphen zur Datenabfrage 41
5.14 Ablaufdiagramm eines Data Stewards bei der Nutzung eines Wissensgraphen zur Sicherstellung der Einhaltung von Compliance-Anforderungen 41
5.15 Ablaufdiagramm eines Data Owners bei der Nutzung eines Wissensgraphen zur Datenbereitstellung 42

6.1 Entity-Relationship-Diagramm des Northwind-Datensatzes 44
6.2 Domänenmodell des Northwind-Datensatzes 45
6.3 Formular zur Erstellung eines Bedarfsknotens im User Interface. 51
6.4 Formular zur Erstellung eines Bedarfsknoten. 53
6.5 Benutzereingabe des User Interface. 54
6.6 Änderung des Status des Bedarfsknoten über User Interface. 55
6.7 Möglichkeit zur Durchsuchung des Wissensgraph. 56
6.8 Sicht auf Graphen aus der Perspektive eines Data Scientists. 58
6.9 Sicht auf Graphen aus der Perspektive eines Data Owner. 58
6.10 Attribute des ausgewählten Knotens *Tofu*. 59

A1 Domänenebene in Neo4j-Browser 71

Tabellenverzeichnis

4.1 Übersicht über die für die verschiedenen Rollen identifizierten Anforderungen . . 22

6.1 Verwendete Technologien im Prototypen 43

7.1 Übersicht über die Umsetzung der Anforderungen im Konzept und Prototyp. . . . 61

Verzeichnis der Listings

6.1 Erstellung der Knoten und Beziehungen des Domänenmodells. 45
6.2 Erstellung der Produktknoten. 46
6.3 Erstellung Beziehungen zwischen Produkten und Kategorien. 46
6.4 Erstellung Beziehung zwischen Instanzen und Domänenmodell. 47
6.5 Erstellung eines Datensatzreferenzknoten. 47
6.6 Beziehung zwischen Domänenebene und Datensatzreferenzebene. 47
6.7 Erstellung des Datenproduktes *AuswertungVerkaufProMitarbeiter*. 48
6.8 Erstellung des Datenproduktes *AuswertungTofu*. 49
6.9 Erstellung operativer Metadaten. 49
6.10 Erstellung technischer Metadaten. 50
6.11 Erstellung geschäftlicher Metadaten. 50
6.12 Cypher-Anfrage zur Erstellung eines Bedarfsknoten. 52
6.13 Cypher-Anfrage um Bedarfsknoten anzuzeigen. 53
6.14 Anzeigen zugewiesener Bedarfsknoten. 54
6.15 Cypher-Anfrage um den Status des Bedarfsknotens zu ändern. 55
6.16 Suche im Wissensgraph. 56
6.17 Überprüfung der Rolle und Filterung der Knoten. 57

Abkürzungsverzeichnis

BI Business Intelligence. 4

DLH Data Lakehouse. 1

DWH Data Warehouse. 1

EKG Enterprise Knowledge Graph. 15

ELT Extract-Load-Transform. 5

ERD Entity-Relationship-Diagramm. 44

ETL Extract-Transform-Load. 6

KI Künstliche Intelligenz. 15

LLM Large Language Model. 64

NLP Natural Language Processing. 16

npm Node Package Manager. 43

OLAP Online Analytical Processing. 4

SSO Single-Sign-On. 54

UID unique Identifier. 34

1 Einleitung

1.1 Hinführung und Problemstellung

Die stetig wachsende Menge verfügbarer Daten und die zunehmende Bedeutung datengetriebener Entscheidungsprozesse in Unternehmen haben den Bedarf an effizienten Methoden zur Datenerhebung und -analyse gesteigert [Göl24]. Mit den Erkenntnisse, die aus diesen Daten gewonnen werden, können Geschäftsprozesse optimiert und Produkte gezielt weiterentwickelt werden, um somit potenzielle Wettbewerbsvorteile zu realisieren [Grö21]. Hierzu müssen Unternehmen jedoch umfangreiche Datenmengen erfassen, verarbeiten und strukturiert verwalten [Grö21].

Um diesen Anforderungen gerecht zu werden, haben sich verschiedene Datenplattformen etabliert, die je nach Architektur und Funktionalität unterschiedliche Ansätze und Vorteile bieten. Data Warehouses (DWHs) bieten eine strukturierte Speicherung und ermöglichen eine zuverlässige Verwaltung relationaler Daten, die in klar definierten Schemata abgelegt werden [GSM14]. Data Lakes hingegen zeichnen sich durch eine flexible Architektur aus, die auch unstrukturierte Datenformate unterstützt, allerdings auf Kosten einer homogenen Struktur und Benutzerfreundlichkeit [Mat17]. Eine neuere Entwicklung ist das sogenannte Data Lakehouse (DLH), das die Stärken von DWHs und Data Lake vereinen soll, um eine effiziente und zugleich flexible Datenverwaltung zu gewährleisten [SGL+24].

Eine zentrale Herausforderung bei der Nutzung von Datenplattformen besteht darin, eine effektive Vermittlung von Daten zwischen Datenproduzenten und Datenkonsumenten zu gewährleisten. Datenproduzenten tragen die Verantwortung für die Bereitstellung und Verwaltung ihrer Daten. Sie müssen sicherstellen, dass die Daten nicht nur in angemessener Verfügbarkeit, Qualität und Struktur vorliegen, sondern auch ausschließlich unter Einhaltung unternehmensinterner und gesetzlicher Vorgaben hinsichtlich Datenschutz und Datensicherheit genutzt werden. Datenkonsumenten hingegen brauchen Zugang zu qualitativ hochwertigen und fachlich relevanten Datensätzen in spezifischen Formaten. Um ihren Datenbedarf effektiv zu decken, benötigen sie Unterstützung sowohl bei der Identifikation geeigneter Datensätze als auch bei der präzisen Formulierung ihrer Anforderungen gegenüber den zuständigen Datenproduzenten. [SL22]

Eine weitere Herausforderung bei der Nutzung von Datenplattformen in Unternehmen besteht darin, die Interaktion zwischen den Datenproduzenten und den Datenkonsumenten zu koordinieren. Die unterschiedlichen Bedürfnisse und Anforderungen dieser beiden Gruppen erfordern eine koordinierte Zusammenarbeit, die insbesondere in großen Unternehmen mit einer Vielzahl von Nutzern und Datensätzen komplex sein kann und daher durch die Datenplattformen erleichtert werden soll.

Moderne Datenplattformen bieten in erster Linie eine technische Basis für die Speicherung und Analyse von Daten. Durch die Integration von Metadatenplattformen, wie etwa Datenkatalogen [PUB23], können jedoch auch organisatorische Anforderungen berücksichtigt werden, wie etwa

die Verwaltung von operationellen Metadaten und die Suche nach Datensätzen innerhalb der Plattform. Allerdings sind die bestehenden Systeme in dieser Hinsicht aktuell noch unzureichend. Erstens greifen sie häufig auf proprietäre Datenmodelle zurück, was zu einer eingeschränkten Flexibilität führt [DSS+20]. Zweitens bieten sie keine holistische Sicht auf die Daten und Prozesse innerhalb der Datenplattform, sodass sie lediglich Teilaspekte abdecken können. Drittens ist die Benutzerfreundlichkeit der Systeme oft unzureichend, wodurch insbesondere Benutzer ohne technische Expertise nicht ausreichend unterstützt werden. Viertens verfügen die Systeme nicht über geeignete Funktionen, um die Kommunikation und Zusammenarbeit zwischen verschiedenen Benutzertypen zu fördern. [PUB23]

1.2 Zielsetzung

Um eine effektive Vermittlung von Daten zwischen Datenproduzenten und Datenkonsumenten zu gewährleistet, bietet sich ein Ansatz an, der auf Wissensgraphen [HBC+21] basiert. Wissensgraphen ermöglichen es, Beziehungen zwischen Entitäten durch Kanten zwischen Knoten darzustellen, was eine flexible und nachvollziehbare Erfassung komplexer Datenstrukturen erlaubt. Darüber hinaus können in Wissensgraphen fachliche Konzepte mithilfe von Ontologien abgebildet und mit Metadaten verknüpft werden. Durch diese Eigenschaften eignen sich Wissensgraphen sowohl für die semantische Beschreibung von Datensätzen auf einer Plattform als auch für die Spezifikation des Datenbedarfs durch Datenkonsumenten. Somit können sie als effektives Mittel zur Vermittlung zwischen Datenproduzenten und -konsumenten fungieren. [YWC+18]

Diese Masterarbeit zielt darauf ab, ein Konzept zur Nutzung von Wissensgraphen zu entwickeln, das den Austausch von Daten zwischen Datenproduzenten und -konsumenten auf Datenplattformen erleichtert. Die entwickelten Konzepte sollen es Datenproduzenten ermöglichen, ihre Daten und deren Eigenschaften klar zu beschreiben, während sie gleichzeitig Datenkonsumenten bei der Suche nach relevanten Daten und der Formulierung ihres Datenbedarfs unterstützen. Zudem sollen Methoden entwickelt werden, welche den Datenkonsumenten zeigen, wie sie die zuständigen Datenproduzenten identifizieren und mit ihnen interagieren können, um benötigte Daten in der gewünschten Form sowie unter Berücksichtigung von Aspekten wie Datenqualität und Governance-Richtlinien zu erhalten. Abschließend werden die Konzepte prototypisch in einem Graphdatenbanksystem implementiert und qualitativ evaluiert.

2 Grundlagen

In diesem Kapitel werden die Grundlagen beschrieben, die zum Verständnis der Arbeit benötigt werden. Zunächst wird das Konzept einer Datenplattformen erörtert, bevor auf die verschiedenen Arten von Datenplattformen eingegangen wird, mit einem besonderen Fokus auf das DLH. Im Anschluss wird ein Zonenreferenzmodell Im Anschluss wird ein Zonenreferenzmodell vorgestellt, bevor der Datenkatalog sowie anschließend die Datenkonsumenten und Datenproduzenten behandelt werden. Darauf folgt ein Abschnitt über Wissensgraphen.

2.1 Datenplattformen

Datenplattformen ermöglichen die Speicherung und Verwaltung von Daten aus verschiedensten Datenquellen für Analysezwecke und bilden damit die Grundlage für die Datenerfassung, Datenverarbeitung und Bereitstellung von Daten für Analyseanwendungen. Diese Plattformen haben sich entwickelt, um den erweiterten Analysebedarf in der Industrie zu decken. [Grö22]

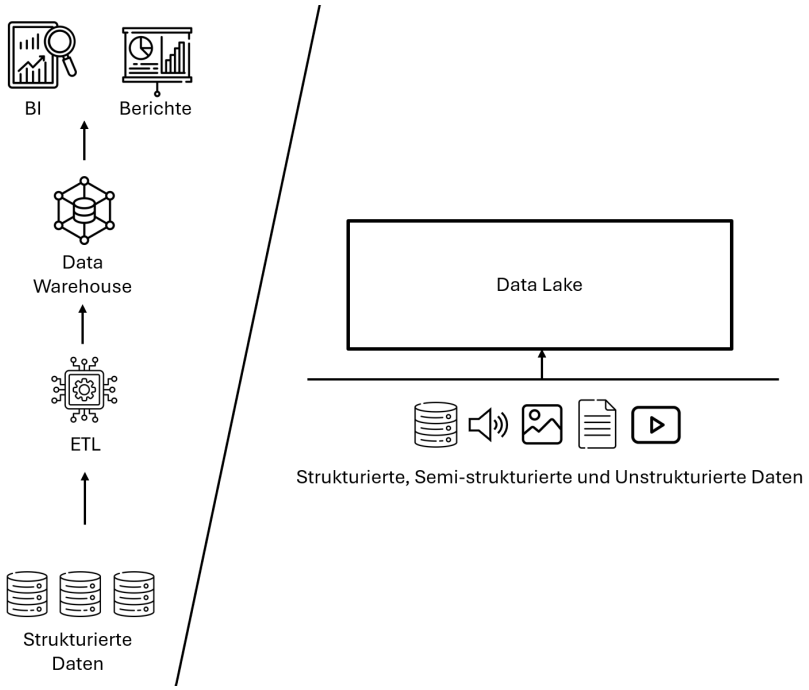


Abbildung 2.1: Schematische Architektur eines Data Warehouses (links) und eines Data Lake (rechts) in Anlehnung an [AGXZ21]

Es gibt verschiedene Arten von Datenplattformen: DWHs für strukturierte Daten, Data Lakes für strukturierte und unstrukturierte Daten und das DLHs. In Abbildung 2.1 ist die Architektur eines DWH und eines DLHs abgebildet.

Ein DWH ist ein spezialisiertes, zentralisiertes System zur Speicherung und Verwaltung **großer** Datenmengen. Es extrahiert und transformiert Daten aus unterschiedlichen Quellen und speichert sie in einer konsolidierten Form, um sie für analytische Abfragen, Business Intelligence (BI) [IBM25b] und Entscheidungsunterstützung bereitzustellen [VZ14]. DWHs sind besonders geeignet, um gezielte Analysefragen zu beantworten, die im Voraus bekannt sind, wie etwa im Rahmen von Reporting und Online Analytical Processing (OLAP)[1]. Im Gegensatz dazu bieten Data Lakes eine flexible, polyglotte Infrastruktur, die aus verschiedenen Datenspeicher- und -verarbeitungssystemen bestehen kann, wie etwa relationale oder NoSQL-Datenbanken, Batch- und Stream-Verarbeitungs-Engines sowie Event Hubs. Diese Flexibilität ermöglicht es Data Lakes, nicht nur vorverarbeitete und aggregierte Daten zu verwalten, sondern auch Rohdaten zu speichern, wie sie von den Datenquellen bereitgestellt werden, die dann schrittweise verarbeitet werden können. [SGL+24] Allerdings geht diese Flexibilität mit einer geringeren Robustheit einher, da Rohdaten bei der Aufnahme kaum anhand eines vordefinierten Schemas validiert werden können. Außerdem führt die Verwendung offener Dateiformate zu einer geringeren Verarbeitungseffizienz im Vergleich zu proprietären DWHs. Die polyglotte Architektur von Data Lakes verhindert zudem eine ganzheitliche Sicht auf die Daten und somit einen einheitlichen Datenzugriff. [SGL+24]

2.1.1 Data Lakehouse

In der Wissenschaft gibt es keine einheitliche Definition von einem DLH. Wo sich jedoch die Definitionen einig sind ist, dass das DLH ein Datenverwaltungssystem ist, das die kostengünstigen Speicher- und Zugriffsfunktionen eines Data Lakes mit den analytischen Fähigkeiten eines DWHs kombiniert [AGXZ21; Dat24a; HZ22; SGL+24]. Daten können aus verschiedenen Quellen extrahiert und ins DLH übertragen werden, wo sie sowohl für strukturierte als auch unstrukturierte Abfragen genutzt werden können, wodurch das System eine flexible und umfassende Datenverarbeitung ermöglicht [AGXZ21; HZ22; SGL+24]. Durch die Kombination beider Arten von Datenplattformen in einem DLH wird es möglich, eine zentrale Datenquelle zu schaffen, die OLAP, Reporting und erweiterte Analysen auf einer einzigen Plattform vereint, was in der obersten Ebene der Abbildung 2.2 zu sehen ist. [AGXZ21]

Ein DLH soll als zentrale Datenbasis fungieren, damit Data Scientists und andere Datenkonsumenten über einen einheitlichen Zugangspunkt auf die benötigten Daten zugreifen können. Dies vermeidet die Notwendigkeit zwischen verschiedenen Datenquellen zu wechseln, die ähnliche, aber potenziell nicht identische, Daten aufgrund von Unterschieden in Datenformaten, Datendomänen oder Erfassungszeiträumen enthalten können. [SGL+24]

[1] OLAP ist eine Technologie, die es ermöglicht, komplexe Abfragen und Analysen großer Datenmengen in Echtzeit durchzuführen. Sie bietet eine multidimensionale Ansicht von Daten, sodass Benutzer schnell und intuitiv auf aggregierte Informationen zugreifen können, um fundierte Entscheidungen zu treffen. [ZH10]

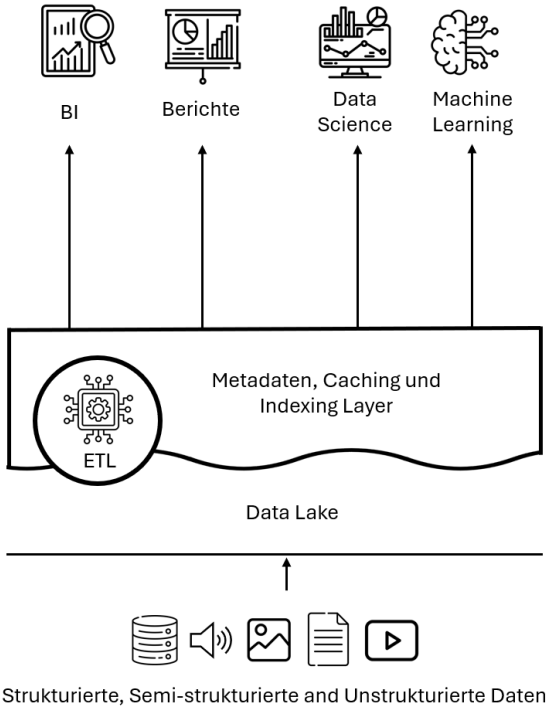


Abbildung 2.2: Schemantische Architektur eines Data Lakehouse in Anlehnung an [AGXZ21]

In einem DLH werden die Daten zunächst mittels des Extract-Load-Transform (ELT)[2]-Verfahrens aus den Datenquellen extrahiert und in den Data Lake geladen. Im Anschluss können die Daten in den verschiedene Zonen des Data Lakes transformiert werden, was in Kapitel 2.1.3 näher beschrieben wird.

Im DLH können Daten sowohl im Rohformat, wie im Data Lake, als auch in aufbereiteter Form, wie im DWH, gespeichert werden. Diese Architektur ermöglicht es, große Mengen an Rohdaten zu speichern und gleichzeitig die Möglichkeit zu haben, diese Daten für spezifische Analysen und Berichte zu verarbeiten und zu strukturieren. [AGXZ21; Dat24a; HZ22; SGL+24]

Rohdaten sind ungeprüfte und unbearbeitete Daten, die direkt von einer Datenquelle stammen. Sie sind in dem Format gespeichert, das von der jeweiligen Quelle bereitgestellt wird. Rohdaten können unstrukturierte Daten, wie Textdokumente oder Bilder, und strukturierte Daten, wie CSV-Dateien oder relationalen Daten, umfassen. [LL20]

Eine robuste Speicherverwaltungsstrategie innerhalb des DLH stellt sicher, dass Daten stets für Benutzer und Anwendungen verfügbar sind. Ein effektives System muss Elastizität, Datenintegrität, Kosteneffizienz, Fehlererkennung und -behebung, Verfügbarkeit, Zugangskontrolle und Haltbarkeit bieten. Diese Eigenschaften gewährleisten eine zuverlässige und sichere Datenspeicherung. [HZ22]

[2]ELT ist ein Datenintegrationsprozess, bei dem Daten aus verschiedenen Quellen extrahiert (Extract), direkt in ein Zielsystem wie ein Data Warehouse oder Data Lake geladen (Load) und anschließend dort transformiert (Transform) werden. [IBM24c]

Im DLH werden Metadaten verwendet, um Daten zu katalogisieren. Metadaten sind, wie in Abbildung 2.2 dargestellt, über dem Data Lake angeordnet. Diese umfassen Informationen über die Struktur der Daten, Herkunft, Qualität und andere relevante Details. [SGL+24] Die Metadaten werden in Datenkatalogen gespeichert und gepflegt, was das Auffinden und die Nutzung der Daten erleichtert [SGL+24].

Die Metadatenschicht kann genutzt werden, um ACID[3]-Transaktionen, Datenqualitätskontrollen, sowie Governance-Funktionen wie Zugriffskontrollen [BGK21] und Audit-Protokollierungen zu implementieren.

Die Metadatenebene nutzt offene Dateiformate wie Parquet[4]-Dateien. [Dat24a] Dadurch können sie erweiterte Verwaltungsfunktionen wie ACID-konforme Transaktionen unterstützen. [AGXZ21; Dat24a] Dies sorgt für Konsistenz, wenn zahlreiche Datenpipelines gleichzeitig auf dieselben Datensätze zugreifen und diese bearbeiten [LAX+20].

2.1.2 Datenkatalog

Ein Datenkatalog ist ein zentrales Verzeichnis, das eine umfassende Sammlung von Metadaten über die Datenbestände einer Organisation enthält [GGH+19; Mic24a; ODO24; PUB23; ZDED17]. Ein wesentlicher Zweck von Datenkatalogen liegt in der Dokumentation des Kontexts, der Bedeutung und des Wertes von Daten, die auf Datenplattformen verwaltet werden. Moderne Datenkataloge sind Metadaten-Repositories [ODO24; ZDED17], welche es den Nutzern ermöglichen, Daten über das gesamte Unternehmen hinweg anzureichern, zu inventarisieren und zu teilen. [Grö22]

Datenkataloge erfassen dabei drei verschiedene Arten von Metadaten: operative, geschäftliche und technische Metadaten. Operative Metadaten umfassen Aspekte der Datenverarbeitung, darunter beispielsweise Zugriffsmuster, Details zu Extract-Transform-Load (ETL)[5] sowie die Herkunft der Daten. Geschäftliche Metadaten sind dafür da, dass Datenkonsumenten die Daten besser verstehen, indem sie Kontext und Bedeutung zu den Daten liefern und somit die Datensätze in den geschäftlichen Kontext einbetten. Dazu gehören Kontext und Verwendung der Metadaten sowie die Data Owner, Ansprechpartner und Tags. Technische Metadaten beschreiben die Struktur des Datensatzes, das Format, die Datenverarbeitung, Datentypen, Spaltennamen, Zugriffsrechte, Datengröße sowie alle Informationen über die Speicherung. [DAM17; DGM+18; IBM24a; PUB23]

Datenkataloge dienen zudem der Erkennung, Beschreibung und Organisation von Datensätzen und ermöglichen es Data Scientists und anderen Datenkonsumenten, relevante Datensätze schnell zu finden und zu verstehen [ZDED17]. Ein Datenkatalog stellt dafür Suchfunktionen bereit, die es ermöglichen, gezielt nach Informationen zu Datensätzen, deren Verarbeitung sowie den verantwortlichen Personen zu suchen. Dies vereinfacht das Auffinden, die Navigation und die Nutzung der Daten. [Clo24; Mic24a; PUB23; Rig24]

[3] Das Akronym ACID steht für Atomicity, Consistency, Isolation und Durability und beschreibt die Regeln und Eigenschaften, die bei Transaktionen in Datenbankmanagementsystemen beachtet werden müssen, um verlässliche und konsistente Daten sowie Systeme zu gewährleisten. [dat25]

[4] Apache Parquet ist ein spaltenorientiertes, Open-Source-Dateiformat, das für die effiziente Speicherung und Abfrage großer Datenmengen entwickelt wurde. Es unterstützt Komprimierungs- und Kodierungsschemata sowie komplexe Datenstrukturen und ist mit zahlreichen Programmiersprachen und Analysetools kompatibel.[Par24]

[5] Extract, Transform und Load bezeichnet den Prozess, bei dem Daten aus verschiedenen Quellen extrahiert, in ein einheitliches Format transformiert und schließlich in ein Zielsystem geladen werden [IBM25c].

Die Speicherung von Metadaten erfolgt meist in tabellarischer Form, wobei Informationen zu den einzelnen Datensätzen in einer strukturierten Tabelle abgelegt werden [ODO24].

Moderne Lösungen für das Metadatenmanagement gehen über die bloße Katalogisierung hinaus und integrieren Geschäftsglossare und Regelverwaltungsfunktionen. In dem Glossar werden geschäftliche und technische Begriffe vereinheitlicht. Damit unterstützen diese bei der Identifizierung von Synergien zwischen Datensätzen, was eine verbesserten Nutzung und Wiederverwendbarkeit der Daten ermöglicht, sowie die Datenkonsistenz verbessert. [Rig24; ZDED17] Regelverwaltungsfunktionen ermöglichen es, Regeln für die Datenqualität und Governance festzulegen und zu verwalten. Dazu gehören das Definieren, Verwalten und Überwachen von Qualitätsstandards und Governance-Richtlinien, die auf Datensätze angewendet werden können. Damit unterstützt ein Datenkatalog unter anderem die Erfüllung von Compliance-Anforderungen, indem er die Relevanz von Datensätzen für spezifische regulatorische Vorgaben analysiert, sie automatisch klassifiziert und für künftige Referenzen kennzeichnet. [Dat24b; IBM24a; Rig24]

Durch das Tagging und Kategorisieren von Datenquellen in einem Datenkatalog können Nutzer relevante Datensätze schneller identifizieren und wichtige Verwendungszwecke hervorheben. Diese Informationen sind sowohl für Datenkonsumenten, als auch für Produzenten zugänglich und erleichtern die gezielte Suche nach Datensätzen. Ein weiterer wichtiger Aspekt ist die Datenklassifizierung, die den systematischen Prozess der Kategorisierung und Strukturierung von Datensätzen anhand festgelegter Kriterien oder Schemata beschreibt. Tagging ist eine spezifische Technik zur Datenklassifizierung, die auch in der Datenverwaltung hilfreich ist. Zum Beispiel bei Compliance- und Sicherheitsaudits können Daten als sensibel oder vertraulich klassifiziert werden, um die entsprechenden Zugriffskontrollen und Archivierungsrichtlinien anzuwenden. [Rag23]

Datenkataloge ermöglichen eine umfassende Analyse der Dateninhalte, -qualität und des Nutzens [ODO24]. Zudem bieten diese auch Funktionen zur Aktivitätsverfolgung von Datenkonsumenten, um Einblicke in die Nutzung und Relevanz der Daten zu gewinnen. Diese Funktionen unterstützen Analysen der Datenherkunft (Data Lineage) [Dat24b] und des Datenflusses [ZDED17]. Die Datenherkunft stellt den Weg dar, den die Daten von der Quelle bis zu ihrem aktuellen Speicherort nehmen, und zeigt alle Änderungen, die auf diesem Weg vorgenommen wurden [IBM24b].

Datenkataloge verfügen zudem oft über Benachrichtigungssysteme, die Benutzer über Änderungen an Datensätzen, Aktualisierungen von Metadaten oder neue Veröffentlichungen informieren. [Nex24; Qli24]

Da nicht allen Nutzern uneingeschränkter Zugang zu sämtlichen Daten gewährt werden kann, stellen Datenkataloge eine Zugriffsverwaltung bereit. Diese ermöglicht es, den Zugriff auf bestimmte Datensätze anzufordern. Nachdem ein Nutzer durch die Suchfunktion einen benötigten Datensatz identifiziert hat, hat er die Option, den Zugriff darauf zu beantragen. Hierfür stehen verschiedene Verfahren zur Verfügung. Beispielsweise können Anfrageformulare genutzt werden. Der Antragsteller muss das Formular ausfüllen und eine Begründung für den Zugriff angeben. Anschließend wird die Anfrage an den Datenproduzenten zur weiteren Bearbeitung weitergeleitet. [Mic24b]

Durch die Bereitstellung benutzerfreundlicher Oberflächen und Funktionen unterstützen Datenkataloge die Datendemokratisierung, indem sie die Datenbestände einer Organisation allen Mitarbeitenden zugänglich machen. Dadurch fördern sie die Datenkompetenz und -nutzung innerhalb des Unternehmens und ermöglichen eine breitere, datengetriebene Entscheidungsfindung. [Mic24a; ZDED17]

Insgesamt fördern Datenkataloge die Datentransparenz und -verfügbarkeit innerhalb von Organisationen und sind damit wichtig für die effektive Verwaltung großer Datenmengen in betrieblichen und analytischen IT-Systemen [Grö22]. Sie unterstützen die automatisierte Extraktion und Anreicherung von Metadaten und spielen eine zentrale Rolle bei der Integration und Nutzung von Daten in Data Lake-Umgebungen [GGH+19].

2.1.3 Zonenreferenzmodell

In diesem Kapitel wird auf die verschiedenen Zonen eingegangen, die ein Data Lake nach Giebler et al. [GGH+20] besitzt, welche in Abbildung 2.3 zu sehen sind. Das Modell hat einen anwendungsfallunabhängigen und einen anwendungsfallabhängigen Teil. Der anwendungsfallunabhängige Teil bewahrt alle ursprünglichen Dateninformationen auf, während der anwendungsfallabhängige Teil Informationsverluste akzeptiert, um spezifische Anwendungen besser zu unterstützen. Jede Zone außer die *Raw Zone* kann weggelassen werden. [GGH+20]

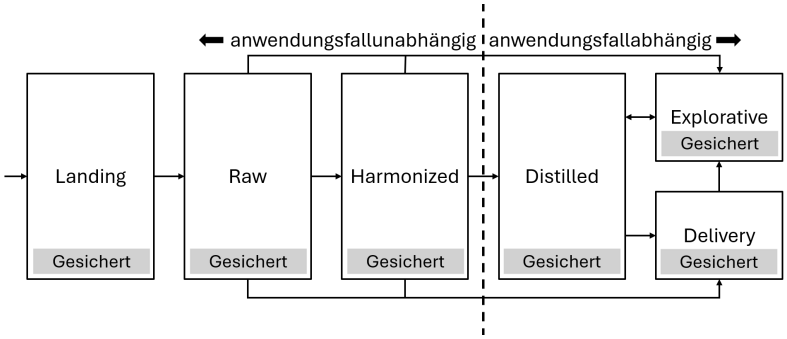


Abbildung 2.3: Zonenreferenzmodell in Anlehnung an [GGH+19]

Wie in Abbildung 2.3 dargestellt, enthalten alle Zonen einen verschlüsselten und gesicherten Teil, der sensible Daten, wie beispielsweise personenbezogene Informationen enthält. Diese Daten wandern von dem geschützten Teil einer Zone zum nächsten und dürfen diesen erst nach Desensibilisierung, wie beispielsweise Anonymisierung, verlassen. [GGH+20]

Die erste Zone ist die *Landing Zone*. Diese wird benötigt, wenn die Anforderungen der aufgenommenen Daten und die der Raw Zone unterschiedlich sind, insbesondere bei großen Datenmengen und hoher Geschwindigkeit. Die Landing Zone ermöglicht die schnelle Aufnahme von Daten und deren Weiterleitung an die Raw Zone. Die aufgenommenen Daten behalten ihre Granularität sowie ihr Quellsystemformat bei. Das Schema der Daten wird nicht geändert, jedoch sind grundlegende Transformationen möglich, wie beispielsweise die Vereinheitlichung von Zeitstempeln. [GGH+20; HORB22]

Die nächste Zone ist die *Raw Zone*. Hier erfolgen die grundlegende Transformationen, wenn es keine Landing Zone gibt. Die Raw Zone speichert Daten dauerhaft, wobei Manipulationen oder Löschungen vermieden werden sollen. In der Praxis erfordert das wachsende Datenvolumen und gesetzliche Vorgaben stellenweise Datenmanipulationen oder -löschungen. Des Weiteren werden die gespeicherten Daten historisiert, wodurch alle Änderungen in den Daten nachvollziehbar bleiben. [GGH+20; HORB22]

Auf die Raw Zone folgt die *Harmonized Zone*. In dieser wird bedarfsabhängig eine Teilmenge der Daten aus der Raw Zone kopiert, ohne dass die Originaldaten gelöscht werden. In der Harmonized Zone werden Daten aus verschiedenen Quellsystemen in ein konsolidiertes Schema integriert und ihre Syntax vereinheitlicht [HORB22]. Die Struktur und Syntax der Daten wird geändert. Die Harmonized Zone zielt darauf ab, eine konsolidierte und vereinheitlichte Sicht auf Unternehmensdaten bereitzustellen. Anstelle eines einzigen, umfassenden Schemas wird die Datenlandschaft in mehrere kleinere Schemata unterteilt, die jeweils spezifische Datenquellen und Kontexte abdecken und schrittweise mit neuen Daten angereichert werden. Diese Teilschemata können miteinander verbunden werden, um größere Teilschemata zu bilden. [GGH+20]

Als nächstes folgt die *Distilled Zone*, welche die erste anwendungsfallspezifische Zone ist. In dieser liegt der Fokus auf der Optimierung der Effizienz nachfolgender Analysen durch gezielte Datenaufbereitung. Daten können hier in ihrer Granularität verändert und ihre Semantik durch komplexe Verarbeitungen angepasst werden. Abhängig von den Anwendungsfällen kann sich auch das Schema geringfügig ändern, wie beispielsweise durch das Hinzufügen von Feldern zur Datenanreicherung. [GGH+20]

Darauf folgt die *Explorative Zone*. Diese Zone ermöglicht es Data Scientists, Daten flexibel zu nutzen und zu untersuchen, ohne feste Datenmerkmale wie in den anderen Zonen zu beachten. Sie können Granularität, Schema, Syntax und Semantik der Daten nach Belieben ändern, wobei sensible Daten strengen Nutzungsregeln unterliegen. Die Zone ist unreguliert und nicht dauerhaft; Ergebnisse können jedoch an die Distilled Zone weitergeleitet werden, bevor die Daten gelöscht werden. Data Scientist haben vollen Zugriff auf diese Zone, können dort Daten transformieren und analysieren sowie Daten aus anderen Zonen einbinden, was eine hohe Flexibilität bei der Datenverarbeitung bietet. [GGH+20]

Die letzte Zone ist die Delivery Zone. Hier werden spezifische Teilmengen von Daten für verschiedene Anwendungen und Nutzerbedürfnisse aufbereitet. Die Dateneigenschaften hängen von den jeweiligen Anwendungsfällen ab und können durch verschiedene Transformationen angepasst werden. Somit können unterschiedliche Transformationen der gleichen Daten durchgeführt werden. Daten in dieser Zone werden verwaltet und dauerhaft gespeichert, solange sie für ihre ursprünglichen Zwecke von Interesse sind. Sie sind für eine breite Nutzerbasis, einschließlich Data Scientists und Domänenexperten, sowie Systeme und Prozesse, zugänglich. Die Daten müssen leicht auffindbar und in verschiedene Analysetools importierbar sein, wobei der Modellierungsansatz je nach Anwendungsfall flexibel gewählt werden kann, wie zum Beispiel mehrdimensionale Modellierung für OLAP oder nicht-normalisierte Tabellen für operative Zwecke. [GGH+20]

2.2 Wissensgraph

Wissensgraphen sind eine Technik, um Wissen übersichtlich und vernetzt darzustellen. Ziel ist es, Informationen über ein Thema so zu sammeln und zu verbinden, dass Zusammenhänge und Beziehungen sichtbar werden. So kann Wissen aus der realen Welt gesammelt und leichter zugänglich gemacht werden. [LCL21; PXNO23] Wissensgraphen stellen einen gerichteten Graph dar, der aus Knoten und Kanten besteht. Ein Knoten repräsentiert dabei eine Entität, die entweder ein reales Objekt oder ein abstraktes Konzept sein kann, während eine Kante die semantische Beziehung zwischen zwei Knoten vermittelt. [YWC+18] Ein abstraktes Konzept in einem Wissensgraphen ist eine Idee, ein Begriff oder ein Prinzip, das nicht direkt physisch existiert oder beobachtbar ist,

sondern nur durch seine Bedeutung oder Funktion in einem bestimmten Kontext verstanden werden kann. Diese Konzepte repräsentieren immaterielle Entitäten, die oft dazu dienen, reale Objekte, Ereignisse oder andere abstrakte Konzepte zu klassifizieren, zu erklären oder in Beziehung zu setzen. [YWC+18] Jede Entität kann durch Merkmale weiter beschrieben und eindeutig identifiziert werden. Die Struktur dieser Entitäten und Beziehungen wird durch Schemata festgelegt, die als Ontologien bezeichnet werden. Diese Ontologien definieren die Typen und Attribute, die in einem Wissensgraph verwendet werden. In Abbildung 2.4 ist ein beispielhafter Wissensgraph abgebildet. Hier wird mit der Entität *Technik* auch ein abstraktes Konzept verwendet. Da die Beziehung zwischen Entitäten nicht notwendigerweise symmetrisch ist, ist die Richtung einer Verbindung wichtig. Daher wird ein Wissensgraph als gerichteter Graph betrachtet, bei dem eine Entität über die Beziehungskante auf eine andere Entität verweist, wie beispielsweise *Bild wird erstellt mit Technik*. [EW16; HBC+21; IBM24b; LCL21; PVGW17; PXNO23]

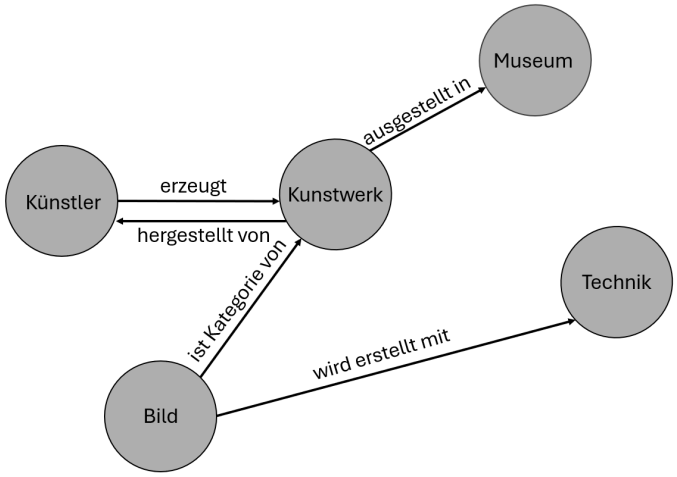


Abbildung 2.4: Beispielhafte Darstellung eines Wissensgraphen

Ein zentrales Element eines Wissensgraphen ist die *Ontologie*, die als Schema für die Beschreibung und Organisation der enthaltenen Informationen dient. In großen Organisationen existieren Daten oft in unterschiedlichen Formaten, wie relationalen Datenbanken, Webseiten, Dokumentationen oder Transaktionsprotokollen. Um diese Daten in einen Wissensgraph zu integrieren, müssen die Daten mittels Data-Lifting in ein neues Format überführt werden. Data-Lifting bedeutet nicht nur die Umwandlung von einem Format in ein anderes, sondern auch die Anhebung der Daten von der reinen Informationsebene in eine maschinenlesbare Wissensebene. Es gibt verschiedene Ansätze zur Durchführung des Data-Liftings, wie Entitätserkennung, Informationsextraktion, Concept Mining und Text Mining, um Wissen aus natürlichen Sprachtexten oder Webseiten zu extrahieren. [PVGW17]

Wissensgraphen schließen die Lücke zwischen der syntaktischen Repräsentation der Daten und deren Bedeutung, indem sie die Geschäftsterminologie der Datenkonsumenten mit den entsprechenden Daten verknüpfen und den Zugriff über diese Begriffe ermöglichen. Dadurch können Anwender auf intuitive Weise auf Daten zugreifen. Zusätzlich bieten Wissensgraphen die Möglichkeit, durch Schlussfolgerungen neues Wissen und Empfehlungen zu generieren. [SL22]

Wissensgraphen bieten ein flexibles und natürliches Datenmodell, das die Integration verschiedener Datenquellen erleichtert. Sie eignen sich besonders gut als Abstraktionsmodell, da sie Daten aus unterschiedlichen Formaten wie relationalen Datenbanken, CSV, XLS, XML, JSON und Textdateien zusammenführen können. Anders als bei relationalen Datenbanken, bei denen das Schema im Voraus definiert werden muss, erlauben Graphen eine dynamische Erweiterung durch das Hinzufügen neuer Knoten und Kanten. Diese Flexibilität macht es einfach, unterschiedliche Graphen zu integrieren, indem Verbindungen zwischen den relevanten Knoten hergestellt werden. Dadurch wird die Datenintegration in Wissensgraphen letztlich zu einem Prozess der Verknüpfung und Vernetzung von Konzepten. [SL22]

3 Verwandte Arbeiten

Aufbauend auf Kapitel 2.1.2 wurde im Rahmen dieser Arbeit eine Literaturrecherche durchgeführt, um die wesentlichen Eigenschaften eines Datenkatalogs zu ermitteln, welche potenziell zur Erfüllung des Ziels beitragen. Anschließend wird aufgezeigt, weshalb Datenkataloge unzureichend für die Erfüllung der Arbeit sind. Anschließend werden durch eine Literaturanalyse weitere Artikel ermittelt, die thematisch mit der Arbeit in Verbindung stehen.

3.1 Datenkatalog

In einem Datenkatalog können Daten auf unterschiedliche Weise strukturiert werden. Wie in Kapitel 2.1.2 beschrieben erfolgt die Speicherung von Metadaten häufig in tabellarischer Form. Moderne Datenkataloge setzen jedoch zunehmend auf graphbasierte Strukturen, um Beziehungen zwischen verschiedenen Datensätzen darzustellen. Diese Graphstruktur visualisiert Verknüpfungen zwischen Datensätzen, Datenquellen und Nutzern sowie deren gegenseitige Abhängigkeiten und ermöglicht eine umfassende Analyse der Datenherkunft und -verflechtung. [He18]

Apache Atlas [Atl24] unterstützt die Modellierung von Metadaten in Form eines graphbasierten Systems, das sich gut eignet, um Entitäten und deren Beziehungen darzustellen, ähnlich wie es in Wissensgraphen der Fall ist. Apache Atlas nutzt die Open-Source-Graphdatenbank JanusGraph [Jan24] zur Suche und Indizierung von Informationen sowie zur Darstellung der Beziehungen zwischen Entitäten im Katalog. Durch die Nutzung von JanusGraph als Backend ermöglicht Apache Atlas die Erfassung und das Durchsuchen komplexer Relationen zwischen Entitäten wie Datenbanken, Tabellen und Spalten. [He18]

Azure Data Catalog [Mic23] und Purview [Mic24d] fokussieren sich vorwiegend auf das Management und die Katalogisierung von Metadaten. Dadurch bieten sie nur grundlegende, graphähnliche Funktionen – beispielsweise Datenherkunftsverfolgung und einfache Verknüpfungen – wodurch die Beziehungen und Abhängigkeiten zwischen Datenquellen lediglich oberflächlich abgebildet werden. Purview unterstützt einige graphenähnliche Funktionen wie Datenherkunftsverfolgung und Metadatenverknüpfungen, um einfache Verbindungen zwischen Datenelementen zu visualisieren [Mic24d]. Für die umfassende Modellierung eines Wissensgraphen, der komplexe Entitäten und vielfältige Relationen detailliert abbildet, sind sie jedoch nicht ausgelegt.

Moderne Datenkataloge integrieren zunehmend Kollaborationsfunktionen, die eine direkte Interaktion zwischen Datenproduzenten und -konsumenten ermöglichen [Nex24]. Diese Funktionen umfassen typischerweise Kommentarfelder zur gemeinsamen Diskussion von Dateninhalten [Bai24; Kli24], Feedbackmechanismen, um Rückmeldungen zur Datenqualität oder Relevanz zu geben [Bai24], sowie Optionen zur Annotation und Dokumentation [Bai24] von Datensätzen, um wichtige Informationen festzuhalten und zu teilen.

Datenkataloge bieten die Möglichkeit, Rollen an Benutzer zu vergeben, um den Zugriff auf Daten und Metadaten zu steuern. So erlaubt beispielsweise das Rollensystem von Apache Atlas, spezifische Berechtigungen für verschiedene Benutzergruppen festzulegen. Dies geschieht durch die Zuweisung von Rollen, die definieren, welche Operationen, wie das Lesen, Schreiben, Aktualisieren oder Löschen von Entitäten, von den jeweiligen Benutzern oder Benutzergruppen durchgeführt werden können. [Apa19; Kat24]

Zusammenfassend lässt sich festhalten, dass Datenkataloge zwar über bestimmte funktionale Eigenschaften verfügen, die zur Unterstützung der Vermittlung von Daten durch einen Wissensgraphen beitragen können, jedoch nicht in der Lage sind, diese Aufgabe in vollem Umfang zu erfüllen.

3.2 Literatur

Um den aktuellen Stand der Forschung zu erfassen, wurde eine komprimierte Literaturanalyse in Anlehnung an Brocke et al. [BSN+09] durchgeführt. Für die Literaturanalyse wurden sechs verschiedene Datenbanken ausgewählt. Dabei handelt es sich um die verlagseigenen Datenbanken ACM (association for computing machinery), digital library, AiSel (Association for Information Systems), IEEExplore (Institute of Electrical and Electronics Engineers), ScienceDirect und SpringerLink sowie die verlagsübergreifende Datenbank Ebscohost.

Die Abfrage der wissenschaftlichen Datenbanken erfolgte mittels einer Stichwortsuche. Es wurden datenbankindividuelle Suchstrings verwendet, was auf die unterschiedlichen Verfügbarkeiten von relevanten Ergebnissen in den einzelnen Datenbanken zurückzuführen ist. Somit mussten manche Datenbanken mehr eingeschränkt werden, um eine überschaubare Menge an Ergebnissen für die anschließende Analyse zu erhalten. Im Anschluss erfolgte eine Rückwärts- und Vorwärtssuche basierend auf den relevanten Artikeln [BSN+09]. Die Rückwärtssuche umfasst die Analyse der Referenzen in den gefundenen Artikeln, während die Vorwärtssuche zusätzliche Quellen berücksichtigt, die den Artikel zitieren [WW02]. Die identifizierte Literatur wurde anschließend auf ihre Relevanz geprüft und entsprechend einbezogen oder ausgeschlossen [BSN+09]. Im Folgenden werden alle relevanten Artikel aufgeführt.

Das Paper von Hoseini et al. [HTQ24] beleuchtet die Rolle von Wissensgraphen und semantischen Modellen im Kontext des semantischen Datenmanagements in Data Lakes. Es legt dabei den Fokus auf die Nutzung dieser Technologien zur Strukturierung und Bereitstellung von Kontext für heterogene Datenquellen. Wissensgraphen dienen hier als zentrale Komponente, um semantische Labels zu erstellen, die Datenquellen mit domänenspezifischen Konzepten verknüpfen. Das semantische Modell erweitert diese Labels durch zusätzliche Beziehungen und Hierarchien, wodurch die Daten innerhalb eines Data Lakes umfassender beschrieben und kontextualisiert werden.

In der Arbeit von Dibowski et al. [DSS+20] wird untersucht, wie semantische Technologien zur Verwaltung von Data Lakes eingesetzt werden können. Dazu wurde ein ‚Semantic Data Catalog' entwickelt, der durch eine semantische Schicht umfassende Beschreibungen der im Data Lake vorhandenen Ressourcen bereitstellt und die semantische Suche ermöglicht. Im Zentrum dieses Semantic Data Catalogs steht ein Wissensgraph, der auf den verfügbaren Metadaten basiert und Instanzen aller Datenressourcen im Data Lake enthält. Die Arbeit legt besonderen Wert auf die Entwicklung und Implementierung dieser semantischen Schicht, um die Auffindbarkeit, Interoperabilität und Wiederverwendbarkeit der Datenressourcen zu verbessern.

Aufbauend auf dem vorherigen Ansatz vertieft ein weiteres Paper von Dibowski et al. [DS21] ein Jahr später, wie speziell Wissensgraphen zur Verwaltung von Data Lakes eingesetzt werden können. Hierbei werden sogenannte semantische Data Lakes eingeführt – eine spezielle Form von Data Lakes, in denen eine semantische Schicht die Daten semantisch anreichert und miteinander verknüpft. Diese semantische Ebene wird als Wissensgraph im ‚Semantic Data Lake Catalog' gespeichert und verwaltet. Der Wissensgraph fungiert dabei als übergeordnete semantische Ebene, die die Bestände des Data Lakes kontextualisiert und durchsuchbar macht.

In der Arbeit von Helal et al. [HHAM21] wird KGLac vorgestellt, ein System, das Metadaten und semantische Informationen von Datensätzen erfasst, um einen Wissensgraphen für Data Lake-Daten zu erstellen. Dieser Wissensgraph ermöglicht es Data Scientists, effizient und skalierbar nach relevanten Daten zu suchen und Beziehungen zwischen Datensätzen zu identifizieren. KGLac soll Unternehmen dabei unterstützen, ihre umfangreichen Datenbestände durch Wissensgraphtechnologien besser zu integrieren und durch gezielte Entdeckungs- und Anreicherungsfunktionen effektiv zu nutzen.

In dem Paper von Helal et al. [Hel21] wird ein System vorgestellt, das die Datensuche in Data Lakes vereinfacht und die Anreicherung der Daten ermöglicht, indem es Wissensgraphen integriert und mit externen Wissensquellen verknüpft. So entsteht ein inhaltlich reichhaltigerer Datenbestand, der durch semantische Strukturierung die effiziente Datenentdeckung und -nutzung unterstützt.

In der Arbeit von Fernandez et al. [FAK+18] wird AURUM vorgestellt, ein System zur Datenentdeckung, das einen Enterprise Knowledge Graph (EKG) erstellt, verwaltet und abfragbar macht, um die Suche nach relevanten Datensätzen zu erleichtern. Der EKG repräsentiert Beziehungen zwischen Datensätzen und unterstützt Analysten dabei, effizient durch diverse Datenquellen zu navigieren und Entdeckungsprobleme in Data Lakes zu lösen. Solche Entdeckungsprobleme umfassen die Identifikation und Extraktion relevanter Datensätze, die zur Erreichung spezifischer Analyseziele beitragen können.

Das Paper von Yahya et al. [YBA21] untersucht die Rolle des Semantic Web und von Wissensgraphen in der Industrie 4.0 [SAP25a]. Es beschreibt, wie Wissensgraphen helfen können, Daten aus unterschiedlichen Quellen zu integrieren, um einen kohärenten Überblick über komplexe Systeme und Prozesse zu bieten. Wissensgraphen fungieren als zentrale Elemente, die nicht nur Daten organisieren, sondern auch deren Bedeutung und Beziehungen untereinander erfassen. Das Paper hebt zudem spezifische Anwendungsfälle hervor und diskutiert die Herausforderungen bei der Implementierung dieser Technologien sowie deren potenzielle Vorteile für die Effizienz und Interoperabilität in industriellen Anwendungen.

Die drei folgenden Arbeiten zeigen unterschiedliche Ansätze zur Nutzung von Wissensgraphen, um die Organisation und Verteilung von Daten auf journalistischen Plattformen zu unterstützen. Der Artikel von Ocaña et al. [OO23] befasst sich mit der Entwicklung einer Architektur speziell für journalistische Wissensplattformen, die Künstliche Intelligenz (KI) und Wissensgraphen integrieren, um die Nachrichtenproduktion zu unterstützen. Das Paper von Rudnik et al. [REF+19] verwendet einen Ereignis-Wissensgraphen, um Nachrichtenartikel zu annotieren und durchsuchbar zu machen. Während der wissenschaftliche Artikel von Opdahl et al. [OAD+22] einen Überblick über die Rolle von Wissensgraphen in der Nachrichtenproduktion und -verbreitung bietet, einschließlich der Anreicherung und Verknüpfung von Inhalten für journalistische Anwendungen.

Das Paper von Kalaycı et al. [KGL+20] präsentiert einen Ansatz zur semantischen Integration von Fertigungsdaten bei Bosch mithilfe virtueller Wissensgraphen. Die erstellte Architektur nutzt eine Ontologie, um heterogene Maschinendaten zu vereinheitlichen und zugänglich zu machen. Mithilfe von Mapping-Regeln werden die Datenquellen mit der Ontologie verknüpft, was eine kohärente, integrierte Sicht auf den gesamten Fertigungsprozess ermöglicht. Dies erleichtert die Formulierung von Abfragen für verschiedene Analysen und Anwendungen, einschließlich Qualitätssicherung, Produktionsoptimierung und Fehlerdiagnose, die direkt auf die semantisch angereicherten Daten zugreifen können.

Der Artikel von Strötgen et al. [STF+19] beschreibt, wie die Bosch Materials Science Knowledge Base einen Wissensgraphen zur semantischen Vermittlung und Organisation von Materialwissenschaftsdaten nutzt. Diese integriert Daten aus verschiedenen Quellen, darunter Datenbanken und wissenschaftliche Publikationen, und nutzt Ontologien zur Strukturierung und Organisation der Informationen. Ein System, das auf Natural Language Processing (NLP) basiert, extrahiert Wissen aus Texten und fügt es dem Wissensgraphen hinzu. Über eine benutzerfreundliche Oberfläche können Nutzer Abfragen mit erweiterten Such- und Analysefunktionen durchführen, einschließlich Inferenz und Herkunftsnachweisen.

In der Literatur existieren weitere Ansätze zur Datenvermittlung, die ohne direkte Verwendung von Wissensgraphen auskommen. So beschreibt der Artikel von Hai et al. [HGQ16] Constance, ein Data-Lake-System zur Verwaltung struktureller und semantischer Metadaten für Rohdaten aus verschiedenen Quellen. Constance ermöglicht eine einheitliche Abfrageverarbeitung und enthält eine spezielle Komponente zur semantischen Zusammenführung von Metadaten, die Ontologien modelliert, Attribute annotiert, Datensätze verknüpft und eine semantische Anreicherung ermöglicht. Das System stellt die Metadaten und ihre Beziehungen grafisch dar, um die Datensuche innerhalb des Data Lakes zu erleichtern.

Das Paper von Ziegler et al. [ZRKM20] beschreibt einen graph-basierten Ansatz zur Verwaltung von Computer-Aided Engineering-Daten in einem Data Lake, um die Datenintegration und -nutzung zu verbessern. Der Ansatz kombiniert die Flexibilität eines Data Lakes für die Speicherung großer, heterogener Datenmengen mit einem graph-basierten Metadatenmanagement, das sowohl die Daten selbst als auch die Beziehungen zwischen ihnen abbildet.

Das Paper von Mami et al. [MGS+19] beschreibt das Konzept eines „Semantic Data Lake“, der durch semantische Technologien eine einheitliche und direkte Abfrage von Daten in Data Lakes ermöglicht. Er nutzt Ontologien und semantische Mapping-Regeln, um heterogene Datenquellen durch eine virtuelle semantische Schicht zu verbinden. Nutzer können Abfragen formulieren, die direkt auf die semantisch angereicherten Daten zugreifen. Das System unterstützt die Aggregation von Informationen aus verschiedenen Quellen und bietet eine konsolidierte Sicht auf die Daten.

Andere Artikel konzentrieren sich auf das Management von Metadaten in Data Lakes und deren Organisation in Form von Graphstrukturen. Dadurch sollen Beziehungen zwischen Daten identifiziert sowie die Nachverfolgbarkeit und Auffindbarkeit verbessert werden. Dazu zählt unter anderem das Paper von Sawadogo et al. [SD21].

Zusammenfassend behandeln sämtliche identifizierten Artikel relevante Aspekte dieser Arbeit – insbesondere den Einsatz von Wissensgraphen zur Optimierung der Datensuche. Dennoch geht die Interaktion zwischen Datenproduzenten und Datenkonsumenten über eine bloße Verbesserung der Auffindbarkeit hinaus, weshalb in der Literaturanalyse keine wissenschaftliche Arbeit gefunden werden konnte, die dieses erweiterte Themenfeld umfassend beleuchtet.

4 Datenkonsumenten und Datenproduzenten

In diesem Kapitel werden zu Beginn die Aufgaben der Datenproduzenten und Datenkonsumenten identifiziert und beschrieben. Diese Analyse stützt sich auf eine gründliche Untersuchung der vorhandenen Literatur, die sowohl aktuelle wissenschaftliche Erkenntnisse, als auch bewährte Methoden berücksichtigt. Daraufhin werden die spezifischen Anforderungen der Datenproduzenten und -konsumenten an die Datenplattform zusammengetragen und eingehend untersucht.

4.1 Aufgaben

Datenproduzenten und -konsumenten stehen für Ressourcen oder Akteure, die Daten entweder bereitstellen oder nutzen [Grö21]. Innerhalb einer Organisation werden Rollen wie Data Steward und Data Owner häufig als Datenproduzenten betrachtet. Diese tragen die Verantwortung für die Konzeption, Organisation und Pflege der Datenstrukturen. Ihr Wissen über komplexe Datenbankschemata ist entscheidend, um einen reibungslosen Datenzugriff zu gewährleisten und eine effektive Erhebung, Bereitstellung und Nutzung der Daten durch andere Beteiligte zu ermöglichen. [SL22]

Datenkonsumenten sind Personen, die Daten für analytische Zwecke benötigen. Dazu gehören unter anderem Data Engineers und Data Scientists. Data Scientists verwenden Daten, um komplexe Muster, Trends und Korrelationen zu identifizieren und somit Geschäftsentscheidungen zu unterstützen. Ihre Analysen dienen der Problemlösung, der Prognoseerstellung und der Generierung von geschäftlichen Erkenntnissen. Data Engineers hingegen konzentrieren sich darauf, eine robuste und skalierbare Dateninfrastruktur zu entwickeln, indem sie Datenpipelines aufbauen, optimieren und den reibungslosen Datenfluss sicherstellen. [SL22]

Probleme treten häufig auf, wenn es zu Missverständnissen zwischen den Datenkonsumenten und Datenproduzenten kommt. Solche Schwierigkeiten können durch das Fehlen einer gemeinsamen Sprache oder die natürlichen Grenzen der menschlichen Kommunikation verursacht werden. Das Ziel ist es, die Nutzer zu befähigen, eigenständig auf Daten zuzugreifen, Abfragen zu erstellen und Berichte zu generieren. Dies soll mit minimaler Unterstützung durch die Datenproduzenten erfolgen, wodurch sowohl Aufwand als auch Zeit gespart und Fehlerquellen verringert werden können. Dies wird auch als *Demokratisierung* von Unternehmensdaten bezeichnet. [SL22]

Die folgenden Abschnitte widmen sich einer detaillierten Beschreibung der einzelnen Rollen, zu denen Data Owner, Data Stewards, Data Engineers und Data Scientists gehören. Darüber hinaus werden die Aufgaben der jeweiligen Rollen basierend auf einer umfassenden Analyse der verfügbaren Literatur zusammengefasst.

4.1.1 Data Owner

Data Owner sind verantwortlich für spezifische Daten innerhalb einer Organisation, wie beispielsweise alle Daten, die sich auf ein bestimmtes Produkt beziehen [Grö21]. Als Datenproduzenten tragen sie aus betriebswirtschaftlicher Perspektive die Verantwortung für die Qualität, Sicherheit und Einhaltung von Compliance-Richtlinien der Daten, welche sie verwalten. Ein zentraler Aspekt ihrer Rolle ist die Schaffung einer klaren und transparenten Struktur für das Dateneigentum, um die ordnungsgemäße Verwaltung und Nutzung der Daten sicherzustellen. [ZZX+23] Ein Data Owner verfügt typischerweise über das nötige Fachwissen und Geschäftsverständnis, um Datensätze effektiv zu verwalten [Grö21; ZZX+23].

Die Aufgaben eines Data Owners werden im Folgenden etwas ausführlicher beschrieben:

DO1 **Datenbereitstellung**: Data Owner sind verantwortlich dafür, ihre Daten in geeigneter Form bereitzustellen. Sie stellen sicher, dass der Zugriff auf die relevanten Datenquellen gewährleistet ist und die Daten jederzeit verfügbar sind. [Ins24; ZZX+23]

DO2 **Implementierung von Datenprozessen**: Der Data Owner setzt Prozesse zur Erfassung, Speicherung, Bereinigung und Transformation der Daten um [Grö21; Ins24].

DO3 **Datenqualitätsmanagement**: Der Data Owner überwacht die Qualität seiner Daten und stellt sicher, dass diese in einem Zustand bereitgestellt werden, der ihre potenzielle Eignung für vielfältige Anwendungsfälle gewährleistet. Dafür wird sichergestellt, dass die Daten korrekt, vollständig, konsistent und aktuell sind. [Grö21; Ins22; Ins24; LL19; ZZX+23]

DO4 **Datensicherheitsmanagement**: Dies umfasst den Schutz der Daten vor unbefugtem Zugriff, Missbrauch und Verlust. Sie verwalten die Zugriffsrechte auf die Daten und sorgen dafür, dass nur autorisierte Benutzer auf diese zugreifen können. [Grö21; Ins22; Ins24; LL19; ZZX+23].

DO5 **Einhaltung von Unternehmensrichtlinien**: Der Data Owner sorgt dafür, dass die Daten in Übereinstimmung mit den Unternehmensrichtlinien genutzt werden [Grö21; Ins22; Ins24; LL19].

DO6 **Konformität der Daten**: Die Daten müssen sowohl den Anforderungen der DSGVO als auch den Qualitätsstandards der Organisation entsprechen [Grö21; Ins22; LL19].

DO7 **Datenarchivierungsmanagement**: Zusätzlich kümmern sich Data Owner um das Datenarchivierungsmanagement, indem sie festlegen, wie Daten klassifiziert, archiviert, aufbewahrt und gelöscht werden. [Ins22; ZZX+23]

DO8 **Definition von Datenanforderungen**: Der Data Owner legt die Anforderungen an die Daten fest, die für den Geschäftsbetrieb erforderlich sind [Ins24; LL19]. Diese Anforderungen müssen definiert werden, um sicherzustellen, dass die Daten den geschäftlichen, qualitativen, rechtlichen und technischen Vorgaben des Unternehmens entsprechen.

D09 **Datennutzung**: Data Owner überwachen die Nutzung der Daten, fördern deren Anwendung und maximieren somit ihren Wert für die Organisation. [Ins22]

4.1.2 Data Steward

Ein Data Steward ist für die Verwaltung von Daten zuständig, sowie für die Sicherstellung von Datenqualität, -sicherheit und -compliance. Er setzt Richtlinien und Verfahren aus geschäftlicher und technischer Sicht um. Dies umfasst die Verantwortung für die Überwachung von Datenprozessen sowie die Sicherstellung der Datenintegrität und die Einhaltung von Standards. [Grö21] Des Weiteren tragen sie als Bindeglied zwischen IT und Geschäftsbereichen dazu bei, qualitativ hochwertige Daten bereitzustellen, die leicht zugänglich sind und den Anforderungen der Data-Governance-Initiativen des Unternehmens entsprechen [Won23]. Durch ihre Arbeit ermöglichen sie es Unternehmen, eine bessere Datenqualität zu erreichen und das Potenzial der Daten auszuschöpfen. [Grö21]

Die Aufgaben eines Data Stewards werden im Folgenden zusammengefasst:

DSt1 **Datenqualität gewährleisten**: Ein Data Steward ist dafür verantwortlich, die Datenqualität sicherzustellen, indem er regelmäßige Überprüfungen, Bereinigungen und Validierungen der Daten durchführt. Dabei sorgt er für die Einhaltung der definierten Qualitätsstandards und behebt proaktiv Probleme wie Inkonsistenzen, Ungenauigkeiten oder unvollständige Datensätze. [Grö21; VBJ20].

DSt2 **Datenmanagement-Standards**: Data Stewards entwickeln und implementieren Richtlinien und Standards für das Datenmanagement [Dat24c; VBJ20; Won23].

DSt3 **Daten-Governance umsetzen**: Data Stewards überwachen Daten-Governance-Richtlinien und setzen diese durch [Dat24c; Won23].

DSt4 **Umsetzung Datensicherheit**: Data Stewards stellen sicher, dass Daten entsprechend der Sicherheitsrichtlinien geschützt werden und der Zugriff nur autorisierten Personen gestattet wird [Dat24c; VBJ20; Won23].

DSt5 **Metadatenmanagement**: Data Stewards sammeln und dokumentieren Metadaten, wie unter anderem Definitionen und Geschäftsregeln, und verwalten diese Daten so, dass sie im besten Interesse aller Beteiligten genutzt werden können. [Dat24c; VBJ20]

DSt6 **Überwachung Datennutzung**: Data Stewards überwachen die Datennutzung, um sicherzustellen, dass die Daten gemäß den definierten Richtlinien, Standards und Compliance-Vorgaben verwendet werden [Grö21; Won23].

4.1.3 Data Engineer

Data Engineers stellen die notwendige Infrastruktur zur Analyse einer großen Menge von Daten bereit, weshalb sie eine zentrale Rolle in der Datenverarbeitung innehaben [Grö21].

Die Aufgaben eines Data Engineers lassen sich wie folgt zusammenfassen:

DE1 **Entwicklung Datenpipelines**: Data Engineers entwickeln robuste Datenpipelines, welche Daten aus verschiedenen Quellen mittels des ETL-Verfahrens importieren [Grö21; Kro22].

DE2 **Integration und Bereinigung von Daten**: Die erstellten Pipelines müssen Daten zunächst integrieren und bereinigen. Somit wird sichergestellt, dass die Daten in einer strukturierten und zugänglichen Form vorliegen, welche für anschließende Geschäftsanalysen genutzt werden können. [Grö21; Kro22; Wut23]

DE3 **Vorbereitung von Daten**: Data Engineers müssen große Mengen neuer Datenpunkte effektiv verarbeiten, da diese oft unstrukturiert und komplex sind. Dies erfordert die Fähigkeit, leistungsstarke ETL-Jobs zu erstellen, um die Daten für nachgelagerte Analysen vorzubereiten. [Kro22; Wut23]

DE4 **Qualitätsstandard sichern**: Data Engineers arbeiten oft mit Daten in allen Formen und verschiedenen Volumen und müssen diese Rohdaten in einen Zustand bringen, der für die geschäftliche Nutzung verwendbar ist. Wenn Daten in einen Data Lake gelangen, sind sie selten in einem verarbeitungsbereiten Zustand. Daher ist es entscheidend, Regeln festzulegen, um die Daten zu behandeln, welche nicht den definierten Qualitätsstandards der DSGVO und des Unternehmens entsprechen. Der Fokus liegt auf der Verarbeitung großer Datenmengen, um die Qualität der Daten auf technischer Ebene zu gewährleisten. [Kro22; Wut23]

DE5 **Datenbereinigung/Data Cleaning**: Die Rolle der Data Engineers umfasst auch die Verantwortung für die Verarbeitung von Datenaktualisierungen, -löschungen und -partitionierungen, insbesondere wenn sie mit speziellen Dateiformaten wie Parquet arbeiten. Insgesamt sorgen Data Engineers dafür, dass die verarbeiteten Daten sauber und konsistent sind und schaffen somit eine einzige, verlässliche Datenbasis. Damit legen sie das Fundament für nachfolgende Analyseprozesse und ermöglichen es, fundierte geschäftliche Entscheidungen zu treffen.[Grö21; Kro22]

DE6 **Datenhandling**: Ein Data Engineer ist für sämtliche Prozesse verantwortlich, die mit der Erstellung, Speicherung und Pflege von Datensätzen zusammenhängen. Ein zentraler Aspekt seiner Tätigkeit ist das Handling, also die Aufbereitung und Weitergabe von Daten. [Wut23]

DE7 **Infrastrukturmanagement** Data Engineers tragen die Verantwortung für die Sicherheit und Zuverlässigkeit der gesamten IT-Infrastruktur. Dazu gehören auch der Aufbau, die Wartung und die kontinuierliche Überwachung der Infrastruktur, um eine stabile und sichere Datenumgebung zu gewährleisten.[Wut23]

4.1.4 Data Scientists

Data Scientists sind verantwortlich für die Analyse und Interpretation strukturierter und unstrukturierter Daten, deren Ergebnisse dann als Grundlage für strategische Entscheidungen dienen können [Cao17; Grö21; KZDB16]. Sie nutzen statistische Methoden, Data Mining und maschinelles Lernen, um Muster zu erkennen, Vorhersagemodelle zu erstellen und Anomalien aufzudecken [Dha13; Grö21]. Angesichts des steigenden Bedarfs an präzisen, automatisierten Entscheidungssystemen sind Fähigkeiten von Data Scientists im Bereich des maschinellen Lernens sowie vertiefte Kenntnisse in Textverarbeitung und Text Mining von Nöten [Dha13].

Im Folgenden werden die wesentlichen Aufgaben eines Data Scientists aufgelistet:

DSc1 **Datenerhebung**: Der Data Scientist ist dafür verantwortlich, die relevanten Daten aus unterschiedlichen Quellen zu identifizieren und zu sammeln, um eine umfassende Grundlage für die Durchführung von Analysen und die Entwicklung von Modellen zu schaffen [Cao17; Grö21; Kla24; KZDB16].

DSc2 **Datenanalyse**: Untersuchung und Auswertung der gesammelten Daten, um Muster, Trends und Beziehungen zu identifizieren. Dies umfasst die Anwendung statistischer Methoden und Datenanalysetools, um Erkenntnisse zu gewinnen und fundierte Entscheidungen zu treffen. [Cao17; Dha13; Grö21; Kla24]

DSc3 **Predictive Analytics**: Nutzung von Modellen und Algorithmen, um zukünftige Ereignisse oder Trends vorherzusagen und konkrete Empfehlungen zur Optimierung von Geschäftsprozessen oder zur Entscheidungsfindung zu geben. [Cao17; Dha13; Kla24]

DSc4 **Visualisierung der Daten**: Erstellen von Grafiken, Diagrammen und Dashboards, um komplexe Daten und Analyseergebnisse anschaulich und verständlich zu präsentieren. [Fiv23; Kla24]

DSc5 **Datenbasis schaffen**: Die Vorbereitung von Daten für die Analyse ist ein grundlegender Schritt, der sicherstellt, dass die Daten in einem geeigneten Format vorliegen und für die Analyse verwendet werden können. Die Aufgabe eines Data Scientists ist es die relevanten Daten zu extrahieren, unwichtige Daten heraus zu filtern und die Daten zu mappen. Des weiteren muss der konvertierte Datensatz in das passende Format gebracht werden. [Cao17; Kla24]

4.2 Anforderungen

In diesem Kapitel werden die Anforderungen der Datenkonsumenten und Datenproduzenten an die Datenplattform zusammengefasst. Diese Anforderungen leiten sich aus den Aufgaben der Datenproduzenten und -konsumenten ab, die im vorherigem Kapitel erarbeitet wurden. Dabei wird gezielt auf die in Kapitel 4.1 aufgeführten Aufgaben verwiesen, indem die jeweiligen Kürzel aus der Auflistung verwendet werden. Des Weiteren werden Anforderungen auf Grundlage der Literatur ergänzt und weiter belegt.

In Tabelle 4.1 sind alle Anforderungen mit einer kurzen Beschreibung aufgelistet. In den hinteren Spalten werden die Anforderungen dabei den Datenkonsumenten, das heißt den Data Ownern (DO) und Data Stewards (DSt) und den Datenproduzenten, also den Data Engineers (DE) und Data Scientists (DSc) zugeordnet. Nachfolgend werden die Anforderungen im Detail beschrieben.

Die Anforderung *A1 (Datenzugriffsverwaltung)* richtet sich an die Datenproduzenten. Sie müssen sicherstellen, dass ihre Daten ausschließlich von autorisierten Datenkonsumenten genutzt werden können. Daher benötigen die Datenproduzenten die Möglichkeit, den Zugriff auf ihre Daten entweder zu genehmigen oder abzulehnen. Dazu soll eine Übersicht bereitgestellt werden, in der die Zugriffsanfragen der Datenkonsumenten aufgelistet sind. Über diese Übersicht können die Datenproduzenten dann entscheiden, ob sie den Zugriff erlauben oder verweigern. Außerdem sollen sie über ausstehende Anfragen benachrichtigt werden. (vgl. DO4, DSt4) [PUB23]

A2 bezieht sich ebenfalls auf die Anforderungen, die an Datenproduzenten gestellt werden. Diese sind verpflichtet, sicherzustellen, dass ihre bereitgestellten Daten den geltenden Compliance-Richtlinien entsprechen. Daher muss eine Möglichkeit geschaffen werden, die es Datenproduzenten ermöglicht, ihre Datensätze vor der Bereitstellung für Datenkonsumenten auf die Einhaltung der Compliance-Vorgaben zu überprüfen. (vgl. DO5, DSt1, DSt3) [PUB23]

Nr.	Anforderung	Kurzbeschreibung	DO	DSt	DE	DSc
A1	Datenzugriffsverwaltung	Übersicht über die Anfragen auf Datensätze, um diese zu genehmigen oder abzulehnen	✓	✓	x	x
A2	Compliance-Anforderung überprüfen	Möglichkeit der Überprüfung, ob Daten Compliance-Anforderungen erfüllen, bevor diese bereitgestellt werden	✓	✓	x	x
A3	Auswertungen über Datenzugriffe	Zugriffsanalysen für Sicherheit, Optimierung und Qualitätsbewertung	✓	✓	x	x
A4	Metadatenmanagement	Sammlung, Verwaltung und Dokumentation von Metadaten	✓	✓	✓	✓
A5	Unterstützung beim Datenupload	System unterstützt beim Upload-Prozess	✓	x	x	x
A6	Datenbenachrichtigung	Informieren über neue Daten	✓	x	✓	✓
A7	Integration in Infrastruktur	System soll plattformunabhängig und einfach integrierbar sein	x	x	✓	x
A8	Datenpipeline-Management	Überblick über Datenpipelines für effiziente Nutzung	x	x	✓	x
A9	Datenpipelines finden	System soll gezielte Suche nach Pipelines ermöglichen	x	x	✓	x
A10	Kommunikation	Austausch zwischen Produzenten und Konsumenten	✓	✓	✓	✓
A11	Verknüpfung von Datensätzen	System ermöglicht kombinierte Datensätze	✓	x	✓	✓
A12	Datenzugriffsanfrage	System ermöglicht es Datenzugriffsanfragen zu stellen	x	x	✓	✓
A13	Intelligente Datensuche	System ermöglicht erweiterte Suche nach relevanten Datensätzen	x	x	✓	✓
A14	Glossar	Einheitliches Glossar verbessert Datensatzsuche und Kommunikation	✓	✓	✓	✓
A15	Rollenverteilung	Rollenverteilung optimiert Datenzugriffe und vereinfacht Prozesse	✓	✓	✓	✓
A16	Datenprodukte	Möglichkeit zur Definition, Erstellung und Vermittlung von Datenprodukten	✓	✓	✓	✓

Tabelle 4.1: Übersicht über die für die verschiedenen Rollen identifizierten Anforderungen

Die Anforderung *A3 (Auswertung über Datenzugriffe)* ist ebenfalls für die Datenproduzenten relevant: Diese benötigen Auswertungen über den Zugriff und die Verwendung ihrer Daten, um die Datensicherheit sicherzustellen, die Datenbereitstellung zu optimieren und wertvolle Einblicke in die Relevanz und Qualität der bereitgestellten Informationen gewinnen zu können. (vgl. DO9, DSt6)

Anforderung *A4*, das *Metadatenmanagement*, ist sowohl für Datenkonsumenten, als auch für Datenproduzenten von Bedeutung. Daher ist ein System erforderlich, das die Sammlung, Verwaltung und Dokumentation aller Arten von Metadaten (siehe Kapitel 2.1.2) ermöglicht. Metadaten sind für Datenproduzenten besonders relevant, da sie eine effiziente Verwaltung und gezielte Bereitstellung von Daten unterstützen. Dabei sind technische Metadaten entscheidend, da sie Informationen zur Speicherung der Daten und zu den Zugriffsrechten enthalten. Für Datenkonsumenten hingegen sind Metadaten wichtig, da sie bei der Identifikation relevanter Datensätze helfen. Besonders geschäftliche Metadaten sind von Relevanz, da sie Kontext und Bedeutung der Daten bereitstellen, was für die Suche nach neuen Datensätzen essenziell ist. Zusätzlich soll das System in der Lage sein, mithilfe von Metadaten Verknüpfungen und Übereinstimmungen zwischen verschiedenen Datensätzen zu identifizieren. Auf diese Weise kann die Suche für Datenkonsumenten optimiert werden, um präzisere oder den Konsumenten noch unentdeckte Datensätze bereitzustellen. (vgl. DSt5, DSc1, DSc5) [PUB23] Darüber hinaus soll das System die Möglichkeit bieten, Datensätze mit zusätzlichen Informationen von Datenkonsumenten anzureichern, beispielsweise wie die Datensätze in Beziehung zueinander stehen. [PUB23]

Anforderung *A5* formuliert den Bedarf von Datenproduzenten, dass das System sie bei dem Hochladen und Bereitstellen von Daten unterstützen soll, um diese Prozesse zu optimieren und zu vereinfachen. Dazu zählt unter Anderem die Automatisierung der Datenübertragung oder auch benutzerfreundliche Schnittstellen. (vgl. DO1, DO8) [PUB23]

Das System soll den Datenproduzenten die Möglichkeit bieten, Datenkonsumenten über die Verfügbarkeit neuer Daten zu informieren, sobald diese hochgeladen wurden, einschließlich spezifischer Informationen zu den neuen Datensätzen. Dies ist mit Anforderung *A6* abgebildet. Das System soll idealerweise die Verknüpfung neu hochgeladener Daten mit bereits bestehenden Datensätzen ermöglichen, um gezielt diejenigen Datenkonsumenten zu informieren, die potenzielles Interesse an diesen neuen Daten haben könnten. (vgl. DO1) [PUB23]

Die Anforderung *A7* an das System besteht darin, dass es nahtlos und sinnvoll in die bestehende IT-Landschaft und Infrastruktur integriert werden kann. Darüber hinaus soll das System eine minimale Abhängigkeit von spezifischen Datenplattformen aufweisen und idealerweise plattformunabhängig für eine breite Nutzerbasis einsetzbar sein. (vgl. DE7)

Anforderung *A8* bezieht sich auf die Aufgabe des Data Engineers, welcher für die Verwaltung von Datenpipelines verantwortlich ist. Hierzu ist es notwendig, dass der Data Engineer einen umfassenden Überblick über die bereits existierenden Datenpipelines besitzt. Dies ermöglicht die Wiederverwendung vorhandener Pipelines, was zur Reduzierung der Systemkomplexität beiträgt und die Redundanz von Datenpipelines minimiert. (vgl. DE1)

Zusätzlich soll das System, wie in *A9* beschrieben, eine Funktionalität bereitstellen, die es ermöglicht, nach bereits bestehenden Datenpipelines zu suchen, um diese effizient wiederverwenden zu können. Optimaler weise soll eine Durchsuchung auf der Grundlage verschiedener Kriterien erfolgen, darunter der Name der Pipeline, die verwendeten Daten, die Aktualität, Schlüsselwörter, Fachbegriffe, Metadaten sowie die Namen der Data Owner. Dadurch können die erforderlichen Datenpipelines gezielt identifiziert und genutzt werden. (vgl. DE1)

Anforderung *A10* umfasst die generelle Idee, dass sich die Datenproduzenten und Datenkonsumenten über das System austauschen können. Ein Austausch ist von Nöten, wenn beispielsweise Fragen zu Nutzungsrechten der Daten aufkommen oder zur Änderung benötigter Daten. Des Weiteren variieren

die Qualitätsanforderungen zwischen den von den Datenproduzenten bereitgestellten Daten und den von den Datenkonsumenten benötigten Daten, weshalb auch hier ein Kommunikationsmittel hilfreich ist. (vgl. DO1, DO3, DO4, DSt1, DSt2, DSt4, DE2, DE3, DE4, DE6 DSc1, DSc5)

A11 beschreibt, dass das System die Möglichkeit bieten muss, Datensätze von verschiedenen Datenproduzenten zu verknüpfen und gebündelt den Datenkonsumenten bereitzustellen. Für viele Analysen sind Datenkonsumenten auf Datensätze unterschiedlicher Data Owner angewiesen. Aus diesem Grund muss das System entweder den Datenproduzenten erlauben die Datensätze vor der Bereitstellung zusammenzuführen, oder den Datenkonsumenten die Möglichkeit bieten, diese nachträglich zu kombinieren. (vgl. DO1, DE1, DE2, DE3, DE5, DSc1, DSc2, DSc3, DSc5)

Aus der vorangegangenen Anforderung sowie aus *A1* kann *A12* abgeleitet werden. *A12* legt fest, dass das System die Möglichkeit bieten soll, den Zugriff auf Daten anzufragen. Dies ist insbesondere dann relevant, wenn Datenkonsumenten aktuell keine Berechtigung besitzen, auf die Daten zuzugreifen, diese jedoch für Analysen benötigen. (vgl. DSc1) [PUB23]

Anforderung *A13* beschreibt den Bedarf an einer umfassenden Suchfunktion im System. Diese soll es Datenkonsumenten ermöglichen, nach Inhalten, Zeitstempeln oder relevanten Begriffen zu suchen, die für ihre Analysen von Bedeutung sind. Das System soll eine intelligente Suche bereitstellen, die nicht nur Datensätze anzeigt, die exakt den eingegebenen Suchbegriff enthalten, sondern auch ähnliche oder thematisch zusammenhängende Datensätze. Dadurch können relevante Datensätze identifiziert werden, die andernfalls unentdeckt geblieben wären. Zusätzlich erleichtert eine benutzerfreundliche Suche, die kein technisches Hintergrundwissen voraussetzt, die Nutzung. (vgl. DSc1, DSc2, DSc3, DSc5) [PUB23]

Des Weiteren soll das System, wie in *A14* beschrieben, für ein einheitliches Verständnis spezifischer Begriffe ein Glossar anbieten, welches die Datenobjekte aus betrieblicher Sicht beschreibt. Damit Datenkonsumenten über die Suche die richtigen Datensätzen angezeigt bekommen, müssen sie die selben Begriffe verwenden, die auch die Datenproduzenten dafür verwendet haben. Ein solches Glossar trägt dazu bei, die Begrifflichkeiten zu vereinheitlichen und eine konsistente Kommunikation zwischen Datenproduzenten und -konsumenten sicherzustellen. [PUB23]

Anforderung *A15* umfasst die Möglichkeiten Nutzern verschiedene Rollen zuzuweisen. Dies bringt den Vorteil einer vereinfachten Rollenverteilung mit sich, da jeder User einer Gruppe zugewiesen werden kann und nicht für jeden individuelle Freigaben eingeholt werden müssen. Somit kann einzelnen Nutzern auch Zugriff auf beispielsweise bestimmte personenbezogene Daten gegeben werden, indem eine Tagverwaltung in die Rollenverteilung mit aufgenommen wird. Das spart sowohl den Datenkonsumenten Zeit, da diese nicht jede Freigabe einzeln einholen müssen, als auch den Datenproduzenten, die nicht jedem Konsumenten einzeln die Freigabe erteilen müssen. (vgl. DO4, DSt4, DE2, DE6, DSc1) [PUB23]

A16 beschreibt die Anforderung, Datenproduzenten und -konsumenten eine Möglichkeit zur Definition, Erstellung und Vermittlung von Datenprodukten bereitzustellen. Datenprodukte sind eine Sammlung von Daten, die für einen bestimmten Zweck aufbereitet und strukturiert wurden [Muc24]. Datenproduzenten sind für die Erstellung und Bereitstellung von Datenprodukten verantwortlich. Datenkonsumenten hingegen nutzen diese, unter anderem für Analysen, und formulieren spezifische Anforderungen, um die für ihre Anwendungsfälle relevanten Daten zu erhalten.

5 Konzeption eines Wissensgraphen

In diesem Kapitel wird, basierend auf den in Kapitel 4.2 identifizierten Anforderungen, ein geeignetes Konzept zur Vermittlung von Daten zwischen Datenproduzenten und Datenkonsumenten auf Datenplattformen entwickelt. Aufgrund der Vielzahl an Anforderungen, beschränkt sich diese Arbeit lediglich auf eine ausgewählte Teilmenge der Anforderungen. Basierend auf der Zielsetzung aus Kapitel 1.2, stehen die Anforderungen im Fokus, die sich mit dem Austausch von Daten zwischen Datenproduzenten und Datenkonsumenten befassen. Dazu gehören die folgenden Anforderungen: *A1*, *A4*, *A6*, *A10*, *A12*, *A13*, *A15* und *A16*.

Zunächst ist Anforderung *A16* von Relevanz, da sie die Definition, Erstellung und Vermittlung von Datenprodukten umfasst – ein zentraler Aspekt, um die effiziente Übertragung von Daten zwischen Datenproduzenten und -konsumenten sicherzustellen. *A10 (Kommunikation)*, spielt eine entscheidende Rolle, da ein effektiver Austausch zwischen den Akteuren unerlässlich ist, insbesondere bei Fragen zur Datenqualität, zu Nutzungsrechten oder zu notwendigen Anpassungen. Ebenso soll sowohl *A1* als auch *A12* berücksichtigt werden. Bei der Datenvermittlung ist eine effiziente Datenzugriffsverwaltung sowie eine klare Regelung für Datenzugriffsanfragen von Bedeutung, um den Bedarf an neuen Daten transparent zu machen. Dabei hat auch *A15*, die Rollenverteilung, von Relevanz. Durch die Zuweisung spezifischer Rollen an die verschiedenen Akteure wird die Datenzugriffsverwaltung vereinfacht und besser strukturiert. Ein weiterer zentraler Aspekt ist die „intelligente Suche“, wie sie in *A13* beschrieben ist. Sie unterstützt Datenkonsumenten bei der Identifikation relevanter Datensätze und erleichtert dadurch die Zusammenarbeit. Ergänzend dazu ist es wichtig, dass Datenkonsumenten über neue Datensätze informiert werden, die ihren Analyseanforderungen entsprechen – ein Punkt, der durch *A6*, die „Datenbenachrichtigung“, abgedeckt wird und daher ebenfalls nicht außer Acht gelassen werden soll. Schließlich spielt ein effektives Metadatenmanagement, wie in *A4* beschrieben eine Schlüsselrolle für beide Personengruppen: Während Datenproduzenten auf Metadaten angewiesen sind, um Daten effizient zu verwalten und bereitzustellen, helfen Metadaten den Datenkonsumenten bei der Identifikation relevanter Datensätze. Dies fördert nicht nur die Nutzung der intelligenten Suche, sondern stärkt auch die Zusammenarbeit zwischen Datenproduzenten und -konsumenten.

5.1 Entwurf eines Wissensgraphen

Um den Wissensgraphen zur Verbesserung der Datenvermittlung einzusetzen, muss zunächst die Struktur und der Aufbau des Graphen erörtert werden. Im Anschluss folgt die Analyse der Data Lineage sowie der Akteure und deren Rollen innerhalb des Wissensgraphen. Abschließend werden das Benachrichtigungssystem, die Suchfunktionalitäten sowie die Aktualisierung des Wissensgraphen behandelt.

5.1.1 Aufbau des Wissensgraphen

Für den Aufbau des Wissensgraphen wurden, wie in Abbildung 5.1 zu erkennen ist, verschiedene Ebenen gewählt. Bei dem aufgezeigten Wissensgraphen handelt es sich um einen Property Graph. Ein Property Graph ist ein Datenmodell, bei dem Knoten und Kanten als grundlegende Elemente dienen, die jeweils mit beliebigen Attributen versehen werden können. Diese flexible Struktur ermöglicht es, neben den reinen Verbindungen auch zusätzliche Metainformationen zu speichern. [RWE15]

Domänenebene: Die oberste Ebene des Wissensgraphen ist die Domänenebene, die zentrale Geschäftsobjekte umfasst. Diese Geschäftsobjekte repräsentieren reale Entitäten, die im geschäftlichen Kontext relevant sind, wie beispielsweise Maschinen, Standorte, Produkte oder Dienstleistungen. Sie fungieren als zentrale Akteure, Prozesse und Objekte im Unternehmensumfeld.

Das Domänenmodell ist eine Darstellung der wesentlichen Konzepte und Beziehungen innerhalb eines bestimmten Anwendungsbereichs. Es dient dazu, die relevanten Objekte, deren Eigenschaften und die Interaktionen zwischen diesen Objekten zu beschreiben. Es dient als abstrakte, konzeptionelle Abbildung eines Arbeits- oder Wissensgebiets. [GP21] Die Detailtiefe des Modells variiert je nach Anwendungsfall und kann individuell angepasst werden. Eine klare Unterscheidung zwischen Stammdaten und Massendaten ist hierbei essenziell; Stammdaten sind Daten, die stabile und unveränderliche Informationen über einen längeren Zeitraum bereitstellen und wiederholt in identischer Form verwendet werden [SAP24]. Massendaten hingegen sind große Mengen an Daten, die typischerweise dynamisch, kurzlebig und prozessbezogen sind [Par23]. Der Wissensgraph fungiert als Datenkatalog, um wichtige Informationen zentral bereitzustellen, ohne die Skalierbarkeit durch eine Überfülle an Daten zu beeinträchtigen.

Datenproduktebene: Darunter folgt die Datenproduktebene, welche die erstellten Datenprodukte enthält. Wie bereits im Anforderungskapitel beschrieben, sind Datenprodukte eine Sammlung von Daten, die für einen bestimmten Zweck aufbereitet wurden [Muc24]. Sie strukturieren die Daten und legen fest, wie diese gesammelt, verarbeitet und analysiert werden, um für den jeweiligen Anwendungsfall relevante Einblicke oder Funktionen bereitzustellen. Durch diese Strukturierung und Kontextualisierung ermöglichen sie eine zielgerichtete Nutzung der Daten.

Datensatzreferenzebene: Die nächste Ebene bildet die Datensatzreferenzebene. Da der Wissensgraph als Datenkatalog fungieren soll, werden die Daten nicht in den Graphen übernommen, sondern nur als Referenzen eingebunden, die auf den Speicherort verweisen und deren Struktur sowie Verfügbarkeit beschreiben. Wie in Abbildung 5.1 zu sehen ist, verweisen die Knoten auf die verschiedenen Zonen des Data Lakes, in welchen die Daten abgelegt sind. Dazwischen befindet sich die semantische Ebene, auf welche in dem folgenden Kapitel 5.1.2 genauer eingegangen wird.

Metadatenebene: Die Metadatenebene ist in der Abbildung 5.1 auf der rechten Seite des Wissensgraphen positioniert, da sie sich über alle anderen Ebenen erstreckt. Diese Ebene fokussiert sich auf spezifische Details zu einzelnen Datensätzen und dient der Organisation verschiedener Metadatenarten, wie sie bereits in Kapitel 2.1.2 beschrieben wurden. Generell können Metadaten auf jeder Ebene auftreten, wobei bestimmte Arten von Metadaten auf spezifischen Ebenen verortet sind. Operative Metadaten, die Informationen über Datenherkunft und -abstammung (Data Lineage) beinhalten, werden primär auf der Datensatzreferenzebene eingeordnet, um die Nachverfolgbarkeit der Daten sicherzustellen. Geschäftliche Metadaten, die den Datenkonsumenten ein besseres Verständnis der Daten ermöglichen sollen, sind der Domänenebene und der Datenproduktebene

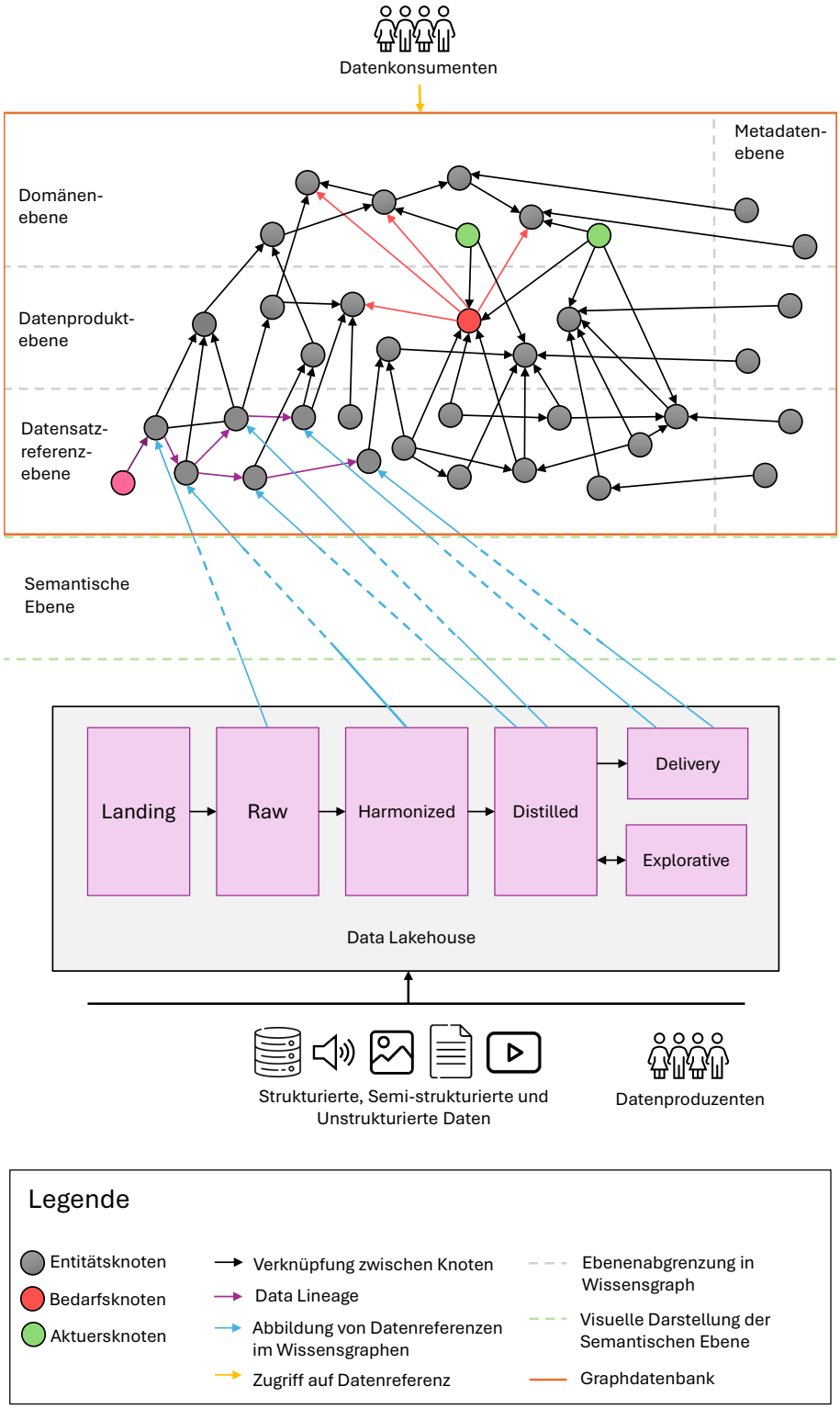


Abbildung 5.1: Aufbau des Wissensgraphen

zuzuordnen. Technische Metadaten, wie Informationen zur Datenstruktur, zum Datenformat oder zur Datenverarbeitung, sind auf der Datensatzreferenzebene zu finden. Die Metadatenebene erstreckt sich somit über alle anderen Ebenen, was in Abbildung 5.1 durch die grau gestrichelte Linie veranschaulicht wird, welche sich durch die Metadatenebene ziehen.

5.1.2 Semantische Ebene

Der Wissensgraph wird nicht im Data Lake gespeichert, sondern in einer Graphdatenbank, was in Abbildung 5.1 durch den orangenen Kasten dargstellt wird. Die Datensätze werden dabei nicht direkt in diese Graphdatenbank geladen, stattdessen werden Referenzknoten erstellt, die auf die entsprechenden Daten im Data Lake verweisen. Der Graph fungiert als eine Art Datenkatalog, der aufzeigt, wo welche Daten zu finden sind. Die Verbindung zwischen den Rohdaten im Data Lake und den strukturierten Daten im Wissensgraphen ist entscheidend, um eine vollständige Data Lineage abzubilden. Eine detaillierte Erläuterung hierzu findet sich in Kapitel 5.1.3.

Zur Anbindung der Daten aus dem Data Lake in eine Wissensgraphstruktur ist ein semantisches Modell erforderlich. Dieses Modell fungiert als Projektionsmechanismus, der die Entitäten und Beziehungen des Wissensgraphen auf die zugrunde liegenden Datenquellen abbildet [HTQ24]. Es stellt eine Vermittlungsschicht dar, die die Datenschicht mit der Wissensschicht verknüpft, wobei die Interaktion zwischen dem Data Lake und der Datensatzreferenzebene wie in Abbildung 5.1 durch die blauen Pfeile dargestellt ist, die vom Data Lake zu den Knoten führen. Auf der untersten Ebene sind die Datensätze im Data Lake lokalisiert, die in unterschiedlichen Formaten vorliegen, wie etwa tabellarische Strukturen oder hierarchische JSON-Daten. Das semantische Modell ist eine Zwischenebene und soll dynamisch gestaltet werden. Diese Flexibilität ist notwendig, wenn Benutzer zusätzliche Datenquellen einbringen, die neue Konzepte und Beziehungen einführen, die im Wissensgraphen noch nicht enthalten sind.

Die Erstellung eines semantischen Modells erfordert eine initiale Analyse der bestehenden Datenquellen und eine präzise Verknüpfung der Datenattribute mit den Konzepten des Graphen. Dieser Prozess kann teilweise automatisiert erfolgen, jedoch bedarf es oft einer manuellen Überprüfung und Verfeinerung, um eine optimale Qualität zu erreichen [HTQ24].

Semantische Modelle werden typischerweise als Graphen modelliert, wobei Knoten Konzepte und Kanten Beziehungen darstellen. Diese Struktur ist flexibel und ermöglicht es, unterschiedliche Datenformate zu integrieren.

5.1.3 Data Lineage

Wie in Kapitel 2.1.3 beschrieben, gibt es nach Giebler et al. [GGH+20] sechs verschiedene Zonen innerhalb eines Data Lakes, welche dieselben Daten unterschiedlich aufbereiten und bereitstellen. Diese sind in Abbildung 5.1 eingezeichnet. In einem Wissensgraphen werden die Daten mit allen vorgenommenen Modifikationen dargestellt. Um dennoch eine Rückverfolgbarkeit zu den Originaldaten sowie deren Quellen sicherzustellen, ist eine Data Lineage wichtig. Die Data Lineage, auch Datenherkunft genannt, beschreibt den Pfad, den die Daten von ihrer Ursprungsquelle bis zu ihrem aktuellen Speicherort zurücklegen, und dokumentiert alle Transformationen und

Änderungen, die auf diesem Weg vorgenommen wurden [IBM24b]. In Abbildung 5.1 ist die Data Lineage als lilafarbenen Pfeile zwischen den Datensatzreferenzknoten dargestellt, einschließlich der Verknüpfung der Datensätze mit den verschiedenen Datenzonen.

Da die Landing Zone als optional betrachtet wird, weil das Schema und das Format des Quellsystems sowie die Granularität der Daten in dieser Zone unverändert bleiben, ist sie für den Wissensgraphen nicht relevant und wird daher nicht berücksichtigt. Somit stellt die Raw Zone die erste relevante Zone dar. Der Datenfluss verläuft demnach vom Datensatz der Raw Zone in die Harmonized Zone. In dieser Zone finden die ersten Änderungen statt, da sowohl die Struktur als auch die Syntax der Daten angepasst werden. Anschließend folgen die anwendungsspezifischen Daten, was zu einer Vielzahl von unterschiedlichen Datensätzen pro Zone führt. Da die Explorative Zone temporär ist (vgl. 2.1.3), wird diese auf Grund des Mehraufwands der zahlreichen erforderlichen Aktualisierungen nicht in dem Wissensgraphen dargestellt.

Um den vollständigen Pfad bis zu den Datenquellen zu erfassen, ist es notwendig, die Datenquellen ebenfalls als Knoten auf der Datensatzebene zu integrieren. Diese fungieren als sogenannte Herkunftsknoten. Diese Datenquellen sind in Abbildung 5.1 pink hervorgehoben. Sie stellen den Ursprung der Daten dar und verweisen demnach auf die Rohdatensätze.

Durch den Einsatz eines Wissensgraphen wird somit eine umfassende und transparente Data Lineage ermöglicht. Ein Wissensgraph modelliert Daten und ihre Beziehungen in einer strukturierten, semantisch angereicherten Form. Dadurch werden Herkunft, Verarbeitungsschritte und Abhängigkeiten von Daten klar und nachvollziehbar dargestellt.

5.1.4 Akteure und Rollen

Um die Interaktion zwischen Datenproduzenten und Datenkonsumenten effektiver zu gestalten, ist die Einbindung von Akteuren und Rollen in den Wissensgraphen essenziell. Akteure interagieren mit den Entitäten, während Rollen definieren, in welcher Funktion diese Interaktion stattfindet und mit welchen Entitäten. Die Einbeziehung dieser Dimensionen erweitert den Wissensgraphen um interaktive Beziehungen, die festlegen, welche Akteure Daten bereitstellen oder konsumieren.

Akteure können zunächst als eigenständige Knoten auf der Domänenebene des Wissensgraphen modelliert werden. Dabei lassen sich spezifische Akteure wie Datenproduzenten und -konsumenten als klar definierte Entitäten einführen. Diese Akteursknoten können durch Attribute wie Name, Funktion oder ihrer Zugehörigkeit zu einer Gruppe näher beschrieben werden. Mit Gruppen sind dabei unternehmensinterne Einheiten gemeint, die den Akteuren auf Basis ihrer jeweiligen Funktion im Unternehmen bestimmte Berechtigungen zuweisen.

Akteure innerhalb eines Wissensgraphen stehen über gerichtete Kanten in Beziehung zu anderen Entitäten. Diese Kanten symbolisieren die Interaktionen eines Akteurs mit verschiedenen Entitäten, beispielsweise Produkten oder Datensätzen. Die Rollen eines Akteurs in diesen Interaktionen können vielfältig sein. Zu den möglichen Rollen zählen etwa: „Nutzt einen Datensatz“, „Besitzt ein Produkt“ oder „Erstellt eine Analyse“. Ein einzelner Akteur kann dabei mehrere Kanten besitzen, die unterschiedliche Beziehungen ausdrücken. So kann ein Akteur gleichzeitig als Owner eines bestimmten Datensatzes auftreten und parallel als Nutzer eines anderen Datensatzes agieren. Akteursknoten, wie beispielsweise die der Data Owner, sind zusätzlich direkt über Kanten mit den Knoten verknüpft, die Datensatzreferenzen repräsentieren. Diese Struktur, die in

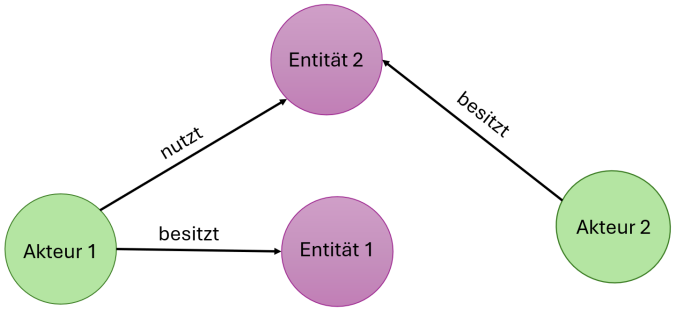


Abbildung 5.2: Darstellung der Akteursknoten und ihrer Interaktion mit Entitätsknoten

Abbildung 5.2 dargestellt ist, ermöglicht es, unmittelbar nachzuvollziehen, welcher Akteur für die Bereitstellung bestimmter Daten verantwortlich ist. In dem Wissensgraphen in Abbildung 5.1 sind die Akteursknoten ebenfalls in grün abgebildet.

Die Zuordnung von Nutzern zu spezifischen Rollen ermöglicht es, die Komplexität des Wissensgraphen für einzelne Rollen gezielt zu reduzieren. Ein Data Scientist benötigt beispielsweise keinen Zugriff auf die Datensatzreferenzebene, weil für ihn der genaue Speicherort der Daten nicht relevant ist, da ihm diese direkt vom Data Owner bereitgestellt werden. Durch diese Rollenzuordnung können unterschiedliche Ebenen des Wissensgraphen für verschiedene Nutzerrollen sichtbar gemacht werden. Dies führt zu einer übersichtlicheren Aufbereitung, die sich auf die für die jeweilige Rolle wesentlichen Informationen beschränkt. Diese gezielte Reduktion der Komplexität steigert nicht nur die Benutzerfreundlichkeit, sondern auch die Effizienz bei der Nutzung des Wissensgraphen.

Außerdem können Rollen genutzt werden, um den Zugriff auf bestimmte Domänen einzuschränken. So muss nicht jeder Data Owner alle Datensatzreferenzknoten aus jeder Domäne einsehen, hinzufügen oder bearbeiten können. Durch diese Rollenverteilung bleibt der Zugriff auf relevante Informationen beschränkt, was gleichzeitig die Übersichtlichkeit erhöht.

5.2 Interaktion mit dem Wissensgraphen

Dieses Kapitel beschreibt verschiedene Möglichkeiten der Interaktion mit dem Wissensgraphen. Dazu gehört die Benachrichtigung, die es ermöglicht, über Aktualisierungen des Wissensgraphen informiert zu bleiben. Zudem wird die Suche im Wissensgraphen behandelt, die Mechanismen zur Informationsabfrage bietet. Abschließend wird die Aktualisierung des Wissensgraphen erläutert, um neue Informationen hinzuzufügen oder bestehende anzupassen.

5.2.1 Benachrichtigungen bei Änderungen des Wissensgraphen

Datenkonsumenten können ihr Interesse an bestimmten Themen innerhalb eines Wissensgraphen auf unterschiedliche Weisen anmerken. Eine Möglichkeit besteht darin, dass die Datenkonsumenten über ihre Knoten im Graphen Interesse an Entitäten ausdrücken. Dies erfolgt durch die Verknüpfung von

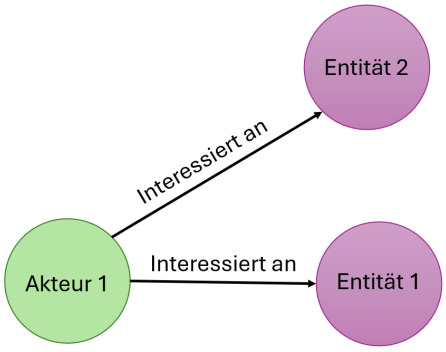


Abbildung 5.3: Darstellung der Verknüpfung eines Akteurs mit einer Entität, an der Interesse besteht

Akteursknoten mit den entsprechenden Entitätsknoten, wobei diese Verbindungen als „interessiert an“ oder „verfolgt“ gekennzeichnet werden können. Abbildung 5.3 veranschaulicht diese Verknüpfung beispielhaft durch zwei Kanten, die einen Akteur mit zwei verschiedenen Entitäten verbinden.

Bei der Integration neuer Daten zu einer Entität im Wissensgraphen können Triggermechanismen aktiviert werden. Diese identifizieren die relevanten Akteurknoten, welche mit den entsprechenden Konsumenten verknüpft sind. Anschließend kann das System automatisierte Benachrichtigungen an die Akteure senden, die für die betreffenden Entitäten zuständig sind, sowie an die Konsumenten, die Interesse an diesen Entitäten bekundet haben.

5.2.2 Suche im Wissensgraph

Um die Vermittlung der Daten zu erleichtern ist es, wie in *A13* beschrieben, ebenfalls wichtig eine *intelligente Suche* zu ermöglichen, um Datenkonsumenten bei der Identifikation relevanter Datensätze zu unterstützen. In einem Wissensgraphen wird die strukturierte Darstellung von Daten und deren Beziehungen genutzt, um komplexe Suchanfragen möglichst präzise zu beantworten und relevante Informationen bereitzustellen. Wissensgraphen speichern Daten in Form von Entitäten und deren Relationen, was es bei der Suche ermöglicht, nicht nur einzelne Begriffe, sondern auch deren Bedeutung und Zusammenhänge zu erfassen. Diese semantische Struktur erlaubt, den Kontext von Anfragen zu verstehen und Antworten zu geben, die über reine Schlagwortübereinstimmungen hinausgehen.

Die Suchfunktion in einem Wissensgraphen erweitert die einfache Rückgabe von Ergebnissen, indem sie dem Nutzer Möglichkeiten zur Verfeinerung seiner Anfrage bietet. So können Vorschläge zu verwandten Themen gemacht werden, die auf den Entitätsbeziehungen basieren und durch Analyse der Knotenverbindungen im Wissensgraphen abgeleitet werden. Auf diese Weise kann der Nutzer durch gezielte Empfehlungen und Ergänzungen zu seiner Anfrage weiterführende Informationen entdecken.

In einem Wissensgraphen können auch Empfehlungsalgorithmen verwendet werden [ZLC+24]. Diese nutzen die Struktur und Verbindungen der Knoten, um Inhalte auf Basis von Ähnlichkeiten und Beziehungen zu empfehlen. Beispielsweise können durch Analyse von Gemeinsamkeiten zwischen Entitäten, Interessen oder Benutzerprofilen Vorschläge generiert werden. Knoten, die

ähnliche Eigenschaften oder Beziehungen zu anderen Knoten haben, können Nutzern als potenziell interessante Themen angezeigt werden. Zwei gängige Methoden der Empfehlung in Wissensgraphen sind die kollaborative Filterung [WHC+19; ZYST19] und die inhaltsbasierte Empfehlung [ZYST19]. Bei der kollaborativen Filterung werden ähnliche Benutzer identifiziert und deren Präferenzen genutzt, um Empfehlungen für andere Benutzer zu generieren. Die inhaltsbasierte Empfehlung hingegen analysiert die Merkmale der vom Nutzer bereits betrachteten Daten und schlägt Inhalte vor, die ähnliche Attribute oder Eigenschaften aufweisen. [WHC+19; WZXG18; ZYST19]

Zusätzlich kann das System Beziehungen zwischen verschiedenen Knoten im Wissensgraphen nutzen, um Datenkonsumenten gezielt auf verwandte Entitäten hinzuweisen. Ein Konsument, der an einer bestimmten Entität interessiert hat, kann beispielsweise auch an den Entitäten interessiert sein, die direkt mit dieser verbunden sind. So können Verbindungen über Kanten zu verwandten Themen führen, die für den Nutzer relevant sein können.

5.2.3 Aktualisierung des Wissensgraph

Die Aktualisierung eines Wissensgraphen kann durch verschiedene Strategien realisiert werden. Eine Möglichkeit ist die periodische Aktualisierung [Bis25], bei der der Wissensgraph in festen Zeitintervallen, beispielsweise täglich oder wöchentlich, aktualisiert wird. Diese Methode eignet sich besonders für Szenarien, in denen die zugrunde liegenden Daten relativ stabil sind und sich nur in größeren Abständen ändern. Durch die Planung fester Aktualisierungszeiten kann zudem die Systemlast gleichmäßig verteilt und eine konsistente Datenbasis sichergestellt werden. Ein Nachteil dieser Strategie besteht jedoch darin, dass zwischen den Aktualisierungszyklen möglicherweise veraltete Informationen im Wissensgraphen verbleiben.

Eine flexiblere Alternative ist die triggerbasierte Aktualisierung [esp25], bei der Änderungen im Wissensgraphen unmittelbar durch bestimmte Ereignisse oder Modifikationen in den zugrunde liegenden Datenquellen angestoßen werden. Diese Methode ist besonders vorteilhaft, wenn Echtzeit- oder nahezu Echtzeit-Daten benötigt werden

Insgesamt soll jedoch eine inkrementelle Synchronisation [Mic24c] angestrebt werden. Dabei werden nicht alle Daten bei jeder Aktualisierung neu geladen, sondern lediglich die veränderten oder neu hinzugefügten Informationen synchronisiert. Dies reduziert die Systemlast und sorgt für eine schnelle Aktualisierung des Wissensgraphen.

5.3 Bedarfsknoten

Im folgenden Abschnitt wird der *Bedarfsknoten*, der in Abbildung 5.1 als roter Knoten mit roten Pfeilen dargestellt ist, eingehend untersucht. Dabei wird besonders die Struktur des Bedarfsknotens detailliert erläutert und dargestellt. Des Weiteren werden die verschiedenen Einsatzmöglichkeiten für Data Scientists, Data Owner und Data Stewards aufgezeigt.

5.3.1 Bedarfsknoten-Ontologie

Zur Darstellung des Bedarfs an Daten können in einem Wissensgraphen Bedarfsknoten spezifiziert werden. Ein solcher Knoten stellt eine Entität dar, die eine spezifische Anfrage eines Datenkonsumenten repräsentiert, wie beispielsweise eine Analyse oder einen Bericht, für den Daten fehlen. Der Bedarfsknoten kann durch zusätzliche Attribute ergänzt werden, die die Art der benötigten Daten charakterisieren, einschließlich Dringlichkeit, Zeitrahmen für die Datenanforderung und spezifische Qualitätsmerkmale.

Es gibt unterschiedliche Arten von Bedarfsknoten, die im Verlauf der Arbeit vorgestellt werden, dazu gehören die Bedarfsknoten auf der Datenproduktebene, der Datensatzreferenzebene, der Metadatenebene und der Domänenebene.

Der allgemeine Bedarfsknoten wird speziell in der Datenproduktebene erstellt. Dies liegt daran, dass ein Bedarfsknoten dazu dient, ein neues Datenprodukt zu erzeugen, wie beispielsweise eine Analyse oder eine Auswertung bestimmter Entitäten. Die Verknüpfung erfolgt primär mit Knoten auf der Domänenebene, es können jedoch auch andere Datenprodukte verknüpft werden, wenn beispielsweise eine bestehende Analyse erweitert werden soll.

Im folgenden wird zunächst eine generelle Ontologie für Bedarfsknoten vorgestellt. Diese ist in Abbildung 5.4 zu sehen. Basierend auf der Ontologie kann ein Bedarfsknoten wie in Abbildung 5.5 aussehen. Im Folgenden werden zunächst die Attribute des Bedarfsknotens erläutert, bevor im Anschluss auf die Relationen eingegangen wird.

```
Bedarfsknoten (Entität: Bedarf)
└── Attribut: Status
└── Attribut:  Name
└── Attribut:  ID / UID
└── Attribut: Beschreibung
└── Attribut: Zeitfenster
└── Attribut: Priorität
└── Attribut: Qualitätsmetriken
└── Beziehung: erstellt → [Entität: Akteur/Datenkonsument]
└── Beziehung: benötigt → [Entität: Domänenentität/(Datenprodukt)]
└── Beziehung: bereitgestellt von → [Entität: Akteur/Datenproduzent]
```

Abbildung 5.4: Struktur eines Bedarfsknotens auf der Datenproduktebene

Attribute

Ein Bedarfsknoten kann verschiedene Attribute enthalten. Attribute sind spezifische Merkmale oder Eigenschaften, die einem bestimmten Bedarfsknoten zugewiesen werden, um ihn zu beschreiben. Der Bedarfsknoten muss die Möglichkeit bieten den aktuellen Status der Datenbereitstellung anzuzeigen. Hierfür kann der Bedarfsknoten eine spezifische Eigenschaft für den Status erhalten. Diese Eigenschaft kann als Attribut `Status` definiert werden, das verschiedene Zustände wie „Offen“, „in Bearbeitung“ oder „erledigt“ annimmt. Im besten Fall wird dafür ein Aktualisierungsmechanismus implementiert, der es ermöglicht den Status des Bedarfsknoten regelmäßig zu aktualisieren. Dies kann

beispielsweise durch einen Trigger oder ein Event-System erfolgen, das reagiert, wenn neue Daten bereitgestellt werden. Diese Trigger können ebenfalls dafür eingesetzt werden, Benachrichtigungen an die Datenkonsumenten zu versenden, wenn alle Daten bereitgestellt wurden.

Zudem gibt es ein Attribut `Name`, welches den Namen des Bedarfsknoten, sowie ein Attribut `ID / UID`, welches die ID beziehungsweise die unique Identifier (UID)[IBM25a] enthält. Darüber hinaus ist das Attribut `Beschreibung` obligatorisch, um eine detaillierte Erläuterung des Bedarfs durch den Akteur zu ermöglichen, der den Bedarfsknoten erstellt hat. Durch diese Beschreibung können Datenproduzenten nachvollziehen, welche Daten relevant sind, und überprüfen, ob entsprechende Knoten auf der Datensatzreferenzebene bereits existieren. Ein weiteres Attribut stellt die `Priorität` dar, welches den Grad der Dringlichkeit des Bedarfs dokumentiert. Zusätzlich dazu ist das Attribut `Qualitätsmetriken` vorgesehen, das die Qualitätsanforderungen des Akteurs an die bereitzustellenden Datensätze spezifiziert.

Relationen

Die Ontologie sieht Relationen des Bedarfsknotens zu anderen Knoten vor, um die kontextuelle Einbettung des Bedarfsknotens zu gewährleisten. Eine zentrale Relation `erstellt` besteht dabei zur Entität des Akteurs, der den Bedarfsknoten erstellt hat und in diesem Zusammenhang als Datenkonsument agiert. Darüber hinaus können spezifische Bedarfe durch die Relation `benötigt` dargestellt werden, die es ermöglicht, den Bedarfsknoten mit anderen Knoten zu verknüpfen, welche die benötigten Ressourcen oder Datensätze repräsentieren. Wie in der Ontologie zu sehen ist können das Domänenknoten oder teilweise auch Datenproduktknoten sein. Datenproduktknoten können angebunden werden, wenn beispielsweise bereits eine ähnliche Analyse existiert, die lediglich angepasst werden muss, um den aktuellen Bedarf zu erfüllen. Falls der Datenproduzent bekannt ist, der den benötigten Datensatz bereitstellt, kann dieser über die `bereitgestellt von`-Relation direkt mit dem Bedarfsknoten verbunden werden. In Abbildung 5.5 ist eine beispielhafte Visualisierung eines Bedarfsknotens dargestellt. Dieser zeigt drei verschiedene Relationen, wobei jede der drei vorgestellten Relationstypen repräsentiert ist. Ein Bedarfsknoten kann jedoch auch zusätzliche Relationen enthalten, wie im nächsten Unterkapitel *Einschränkungen* näher erläutert wird.

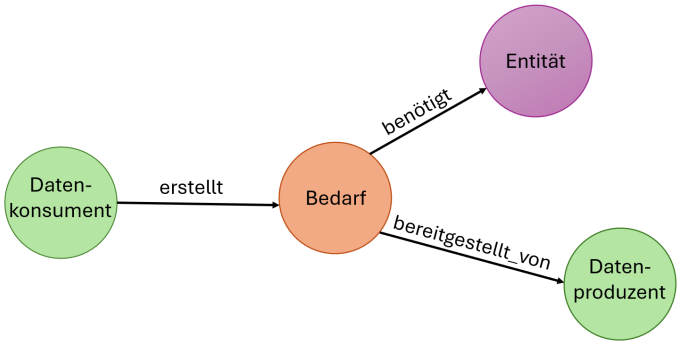


Abbildung 5.5: Visualisierung der Struktur eines Bedarfsknotens auf Datenproduktebene

Einschränkungen

In einer Ontologie lassen sich Einschränkungen definieren, um die Struktur und Gültigkeit der Relationen und Attribute zu präzisieren. Solche Einschränkungen legen etwa fest, wie viele Relationen ein Knoten zu anderen Knoten eingehen darf, oder welche Attribute und Relationen zwingend erforderlich sind, um die Konsistenz des Modells sicherzustellen. Alle Attribute eines Bedarfsknotens sind obligatorisch, sodass jeder Bedarfsknoten umfassend und eindeutig beschrieben ist. Bei den Relationen gelten jedoch spezifische Anforderungen: Die `benötigt`-Relation muss mindestens einmal vorhanden sein, da ein Bedarfsknoten sonst keine funktionale Relevanz hätte. Der Bedarfsknoten hat ohne diese keine Verbindung zu den angeforderten Datenprodukten und ist isoliert. Auch die `angefordert von`-Relation ist zwingend und muss mindestens einmal vorliegen, um anzuzeigen, welcher Akteur, in diesem Fall Datenkonsument, den Bedarf formuliert hat. Es ist jedoch zulässig, dass mehrere Akteure als Datenkonsumenten angegeben werden, falls der Bedarf von mehreren Personen initiiert wird. Die `bereitgestellt von`-Relation ist die einzige, die optional ist. Falls der Bedarfsknoten-Ersteller nicht weiß, wer den benötigten Datensatz bereitstellen wird, kann der Bedarfsknoten auch ohne diese Relation angelegt werden.

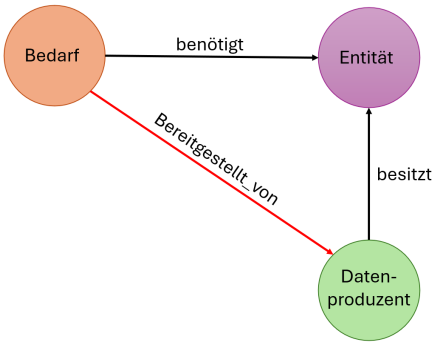


Abbildung 5.6: Automatische Erkennung des Datenproduzenten bei Erstellung des Bedarfsknoten

Abbildung 5.6 zeigt diesen Fall durch eine schwarze Verbindungslinie, die auf den Entitätsknoten verweist, aber noch keinen konkreten Datenproduzenten nennt. Da die Ontologie auf einem Wissensgraphen basiert, soll jedoch ein Mechanismus für Reasoning [RMXS19] vorhanden sein, der die Relation `bereitgestellt von` automatisch schlussfolgern kann. Hierfür können indirekte Relationen genutzt werden: Der Bedarfsknoten ist durch die Relation `benötigt` mit einem Datenprodukt verknüpft. Dieses Datenprodukt wiederum enthält die Relation `besitzt`, welche auf den Data Owner verweist, der in der Ontologie als Akteursknoten dargestellt wird. Durch die Anwendung von Reasoning auf diese beiden Relationen kann im Wissensgraphen die dritte Relation, `bereitgestellt von`, logisch erschlossen und automatisch erstellt werden. Diese Relation wird in Abbildung 5.6 durch eine rote Verbindungslinie angezeigt. Bei der initialen Erstellung eines Bedarfsknotens ist es somit nicht zwingend erforderlich, die Relation `bereitgestellt von` manuell zu definieren.

5.3.2 Weitere Arten von Bedarfsknoten

In dem vorherigen Abschnitt wurde auf den Bedarfsknoten auf Datenproduktebene eingegangen. Ein Bedarfsknoten kann jedoch ebenfalls auf anderen Ebenen eingefügt werden. Der Aufbau und die Erstellung eines Bedarfsknoten bleibt die Gleiche auf allen Ebenen des Wissensgraphen, jedoch ändert sich die Ontologie und die Verwendung dieser.

Bedarfsknoten auf der Datensatzreferenzebene

Bedarfsknoten sind auch auf der Datensatzreferenzebene möglich. Im vorherigen Abschnitt wurde erläutert, wie Anfragen auf der Datenproduktebene formuliert werden können, bei denen Datenproduzenten aufgefordert werden, die benötigten Daten bereitzustellen. Da es für die meisten Entitäten eine Vielzahl von Daten in unterschiedlichen Verarbeitungsformen und Formaten gibt, sind oft mehrere Data Owner für verschiedene Datensätze derselben Entität verantwortlich.

Auf der Datenproduktebene erfolgt die Anfrage jedoch allgemein nach Daten, was bedeutet, dass der Verantwortliche der Entität lediglich darüber informiert wird, dass Daten zu seiner Entität bereitgestellt werden sollen. In der Folge muss dieser auf der Datensatzreferenzebene ermitteln, wo sich die erforderlichen Daten befinden und wer diese bereitstellen kann. Daher muss er möglicherweise verschiedene Datensätze, die unterschiedlichen Data Ownern zugeordnet sind, kombinieren, bevor er sie gebündelt zur Verfügung stellen kann. Um diesen Prozess zu unterstützen, kann er einen Bedarfsknoten auf der Datensatzreferenzebene erstellen. In diesem Fall ist die Ontologie dieselbe wie die in Kapitel 5.3.1 beschriebene.

```
Bedarfsknoten (Entität: Bedarf)
└── Attribut: Status
└── Attribut:  Name
└── Attribut:  ID / UID
└── Attribut: Beschreibung
└── Attribut: Zeitfenster
└── Attribut: Priorität
└── Beziehung: erstellt → [Entität: Akteur/Data Steward]
└── Beziehung: benötigt → [Entität: Datensatz]
└── Beziehung: bereitgestellt von → [Entität: Akteur/Data Owner]
```

Abbildung 5.7: Struktur eines Bedarfsknotens auf der Datensatzreferenzebene im Kontext der Datenqualität

Neben der Datensammlung lassen sich Bedarfsknoten auf der Datensatzreferenzebene auch gezielt zur Sicherstellung und Verbesserung der Datenqualität einsetzen. Solche Knoten können den Bedarf an Maßnahmen zur Qualitätssteigerung markieren, wie etwa die Bereinigung fehlerhafter oder inkonsistenter Datensätze. Data Stewards erstellen diese Bedarfsknoten, wenn sie Datensätze identifizieren, die beispielsweise unternehmensinterne Richtlinien nicht erfüllen oder anderen Qualitätsanforderungen nicht gerecht werden.

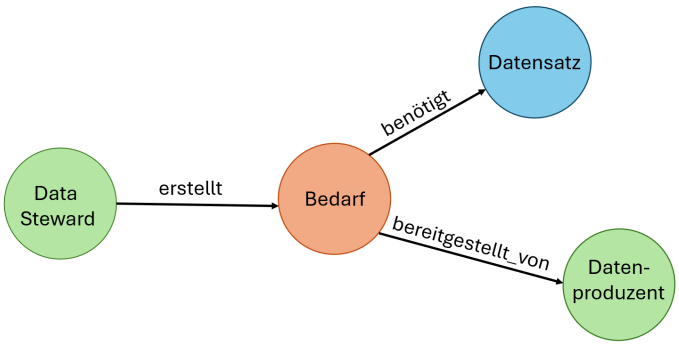


Abbildung 5.8: Aufbau eines Bedarfsknoten auf der Datensatzreferenzebene

In diesem Fall benötigt der Bedarfsknoten eine spezialisierte Ontologie, da der Fokus nun auf der Sicherstellung der Datenqualität und der Einhaltung von Richtlinien liegt, anstatt auf der reinen Datenbereitstellung. Die meisten Attribute des Bedarfsknotens können bestehen bleiben, wie in Abbildung 5.7 dargestellt. Das Attribut `Beschreibung` wird weiterhin genutzt, allerdings mit dem Ziel, den spezifischen Richtlinienverstoß oder die Qualitätsanforderung zu beschreiben, die im Datensatz erfüllt sein müssen. Auch bei den Beziehungen ergeben sich Änderungen. Die Beziehung `Angefordert von` ist auf den Data Steward festgelegt, der den Bedarfsknoten initiiert. Die Beziehung `benötigt` verweist auf den relevanten Datensatz oder eine entsprechende Datensatzreferenz, die überprüft oder angepasst werden muss. Schließlich wird die Beziehung `bereitgestellt von` einem Data Owner zugeordnet, der für die Bereitstellung und die Qualitätskontrolle des Datensatzes verantwortlich ist. In Abbildung 5.8 ist ein solcher Bedarfsknoten abgebildet.

Bedarfsknoten auf der Metadatenebene

Bedarfsknoten können auch auf der Metadatenebene eingesetzt werden, um spezifische Anforderungen an die Qualität und Vollständigkeit der Metadaten zu dokumentieren. Solche Bedarfsknoten können zum Beispiel die Notwendigkeit hervorheben, zusätzliche Metadatenfelder zu definieren oder vorhandene Metadaten zu aktualisieren, um die Datenqualität und deren Nutzbarkeit zu steigern. Ein Bedarfsknoten kann dabei signalisieren, dass erweiterte Angaben zur Herkunft oder zur Nutzungshistorie eines Datensatzes erforderlich sind, um dessen Transparenz und Nachvollziehbarkeit für Anwender zu verbessern. In Abbildung 5.9 ist ein solcher Bedarfsknoten abgebildet.

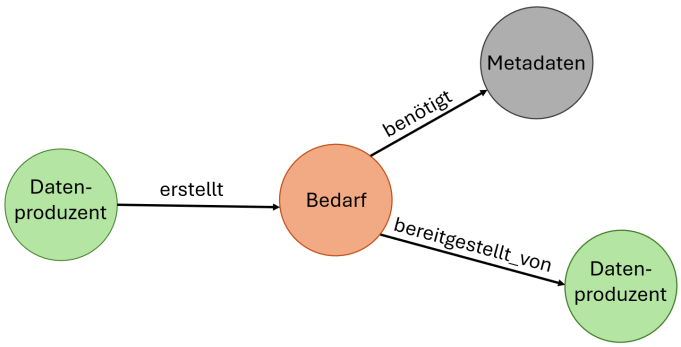


Abbildung 5.9: Aufbau eines Bedarfsknoten auf der Metadatenebene

Ein solcher Bedarfsknoten kann von Datenproduzenten erstellt werden, um spezifische Anforderungen an die Metadaten festzuhalten. Ein Data Steward kann dabei bestimmte Metadatenfelder verlangen, die bereitgestellt werden müssen, um sicherzustellen, dass Compliance-Anforderungen und Unternehmensrichtlinien erfüllt sind und eine Data Lineage aufgezeichnet ist. Die Ontologie eines solchen Bedarfsknotens ist in Abbildung 5.10 zu sehen.

```
Bedarfsknoten (Entität: Bedarf)
└── Attribut: Status
└── Attribut:  Name
└── Attribut:  ID / UID
└── Attribut: Beschreibung
└── Attribut: Zeitfenster
└── Attribut: Priorität
└── Attribut: Qualitätsmetriken
└── Beziehung: erstellt → [Entität: Akteur/Data Steward/Datenproduzent]
└── Beziehung: benötigt → [Entität: Metadaten]
└── Beziehung: bereitgestellt von → [Entität: Akteur/Data Owner]
```

Abbildung 5.10: Struktur eines Bedarfsknotens auf der Metadatenebene zur Metadatenanfrage

Bedarfsknoten auf der Domänenebene

Auch auf der Domänenebene können Bedarfsknoten erstellt werden, allerdings sind diese abstrakter als auf den anderen Ebenen. Da sich auf dieser Ebene die Entitäten und das Domänenmodell befinden, haben die erstellten Bedarfsknoten hier eine andere Funktion. Das Domänenmodell, wie in Kapitel 5.1.1 beschrieben, ist eine grafische Darstellung der Entitäten innerhalb einer Domäne, ihrer Eigenschaften und der Beziehungen zwischen ihnen. Dieses Modell wird initial erstellt, ist sehr stabil und verändert sich nur selten im Laufe der Zeit.

```
Bedarfsknoten (Entität: Bedarf)
└── Attribut: Status
└── Attribut:  Name
└── Attribut:  ID / UID
└── Attribut: Beschreibung
└── Attribut: Zeitfenster
└── Attribut: Priorität
└── Beziehung: erstellt → [Entität: Datenproduzent/Datenkonsument]
└── Beziehung: verbindet → [Entität: Domänenentität]
└── Beziehung: bereitgestellt von → [Entität: Domänenverantwortliche]
```

Abbildung 5.11: Struktur eines Bedarfsknoten auf der Domänenebene

Ändern sich jedoch Geschäftsanforderungen, beispielsweise durch neue Prozesse oder Änderungen in der Domäne, müssen diese Anpassungen auch im Wissensgraphen abgebildet werden. Hierzu kann ein Bedarfsknoten auf der Domänenebene angelegt werden. Der Bedarfsknoten wird auf der Domänenebene ebenfalls mit Entitäten verbunden, jedoch nicht, um einen Bedarf an bestehenden

Entitäten auszudrücken. Stattdessen signalisiert die Verbindung, dass eine neue Entität in das Domänenmodell aufgenommen werden soll. Die Verbindungen zeigen, wo und wie die neue Entität in das bestehende Domänenmodell integriert werden soll. Diese Beziehung wird in Abbildung 5.11 durch die Relation `verbindet` dargestellt. Über das Attribut `Beschreibung` kann die genaue Art der Beziehung zwischen den Knoten definiert werden.

Die Ontologie der Bedarfsknoten auf der Domänenebene ähnelt derjenigen der anderen Bedarfsknoten, weist jedoch zwei wesentliche Unterschiede auf, wie in Abbildung 5.11 zu sehen ist. Erstens wird eine `verbindet`-Relation erstellt, um die Integration neuer Entitäten in das Domänenmodell darzustellen. Zweitens sind keine Qualitätsmetriken enthalten, da es in diesem Kontext keine Daten zur Bereitstellung gibt. Eine Visualisierung des Bedarfsknotens ist in Abbildung 5.12 zu sehen.

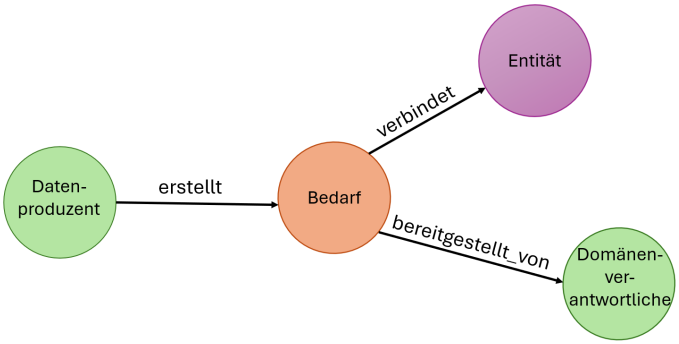


Abbildung 5.12: Aufbau eines Bedarfsknoten auf der Domänenebene

Es ist möglich, bei der Erstellung des Bedarfsknotens benutzerdefinierte Namen für die Relationen zu vergeben. Der Akteur, der den Bedarfsknoten anlegt, kann so direkt den passenden Namen der Verbindung hinterlegen, der die Art der Beziehung in dem Domänenmodell widerspiegelt.

Als verantwortlicher Akteur für die Bearbeitung des Bedarfsknotens soll der Autor des Domänenmodells oder der Domänenverantwortliche hinterlegt werden. Dieser entscheidet dann, ob es sinnvoll ist, das Domänenmodell an der vorgeschlagenen Stelle zu ergänzen.

5.3.3 Verwendung des Bedarfsknotens

Durch die Beziehung, die zwischen dem Bedarfsknoten und dem Datenproduzenten hergestellt wird, wird der Datenproduzent oder Domänenverantwortliche automatisch über die Erstellung eines Bedarfsknotens und die zugehörige Datenanfrage informiert. Dazu können die selben Mechanismen genutzt werden, die in Abschnitt 5.2.1 beschrieben wurden. Anschließend kann er die Attribute des Bedarfsknotens einsehen und erhält dadurch alle relevanten Informationen, wie beispielsweise Qualitätsmetriken. Zudem hat er die Möglichkeit, auch die anderen beteiligten Datenproduzenten einzusehen.

Datenproduzenten können zunächst im Wissensgraphen prüfen, ob die benötigten Daten bereits vorhanden sind. So lässt sich feststellen, ob der Data Owner diese Daten möglicherweise bereits für andere Datenkonsumenten bereitgestellt hat. Sind die benötigten Daten vorhanden, kann der Data Owner sie ohne weitere Anpassung bereitstellen. Falls die Daten jedoch noch nicht existieren, muss der Data Owner sie entsprechend transformieren, um den spezifischen Bedarf des Datenkonsumenten zu erfüllen.

Nachdem der Datenproduzent die benötigten Daten bereitgestellt hat, muss er den Status des Bedarfsknotens aktualisieren. Dadurch wird der Datenkonsument informiert, dass ein Teil der angeforderten Daten verfügbar ist.

5.3.4 Lebenszyklus von Bedarfsknoten

Sobald die Datenproduzenten alle erforderlichen Daten bereitgestellt haben, gilt der Bedarfsknoten als abgeschlossen. Diese abgeschlossenen Bedarfsknoten verbleiben zunächst im Wissensgraphen, um die Nachvollziehbarkeit zu gewährleisten und anderen Interessenten den Zugriff auf das Datenprodukt zu ermöglichen.

Im weiteren Verlauf gibt es verschiedene Möglichkeiten, um mit abgeschlossenen Bedarfsknoten zu verfahren. Grundsätzlich soll ein Bedarfsknoten für eine bestimmte Zeitspanne bestehen bleiben. Nach Ablauf dieser Zeit kann der Bedarfsknoten entweder gelöscht oder in einen dauerhaften Knoten umgewandelt werden, abhängig vom Inhalt des Bedarfsknotens.

Da das Endergebnis eines Bedarfsknotens auf der Datenproduktebene ein fertiges Datenprodukt ist, kann es sinnvoll sein, den Bedarfsknoten in einen Datenproduktknoten umzuwandeln. Diese Möglichkeit besteht auch auf den anderen Ebenen. Wenn auf der Metadatenebene beispielsweise Metadaten fehlen, sollen die bereitgestellten Metadaten über den Bedarfsknoten in Metadatenknoten umgewandelt werden, da diese auch für andere Akteure von Bedeutung sein können. Ebenso sollen auf der Domänenebene die Bedarfsknoten in normale Domänenknoten überführt werden, um das Domänenmodell zu erweitern. Bedarfsknoten, die von Data Stewards erstellt wurden, sollen nicht in reguläre Knoten überführt werden, da sie ausschließlich dazu dienen, fehlerhafte Knoten zu identifizieren und eine Überarbeitung der entsprechenden Daten anzustoßen.

5.4 Interaktion von Akteuren mit dem Wissensgraph

In diesem Kapitel wird ein Konzept vorgestellt, das die Interaktion der verschiedenen Akteure mit dem Wissensgraphen beschreibt. Zur Veranschaulichung werden dabei drei Diagramme herangezogen. Jedes Diagramm beleuchtet den Prozess aus einer anderen Perspektive: Abbildung 5.13 aus Sicht des Data Scientists, Abbildung 5.15 aus dem Blickwinkel des Data Owners und Abbildung 5.14 aus Perspektive des Data Stewards. Obwohl der Data Engineer eine unterstützende Rolle bei der Datenbereitstellung übernimmt, ist er für die in diesem Kapitel behandelten Konzepte nicht relevant und wird daher nicht weiter berücksichtigt. Wie in den Aufgaben des Data Engineers beschrieben, stellt dieser die notwendige Infrastruktur für die Analyse von Big Data [SAP25b] zur Verfügung. Dieser Aspekt wurde jedoch in den zu Beginn des Kapitels behandelten Anforderungen nicht einbezogen. Daher kann der Data Engineer in der weiteren Diskussion außen vor gelassen werden. Die Ablaufdiagramme wurden in Anlehnung an den UML-Standard als Aktivitätsdiagramme [Luc24] erstellt.

In Abbildung 5.13 ist das Ablaufdiagramm eines Data Scientists dargestellt, das zwei Startknoten umfasst. Der erste Knoten wird aktiviert, wenn der Data Scientist neue Daten benötigt, während der zweite Knoten zum Einsatz kommt, sobald eine Benachrichtigung über die Verfügbarkeit neuer Daten eingeht, was in Kapitel 5.2.1 beschrieben wurde. Im weiteren Verlauf hat der Data Scientist die Möglichkeit, im Wissensgraphen nach den benötigten Knoten zu suchen. Wenn er einen relevanten

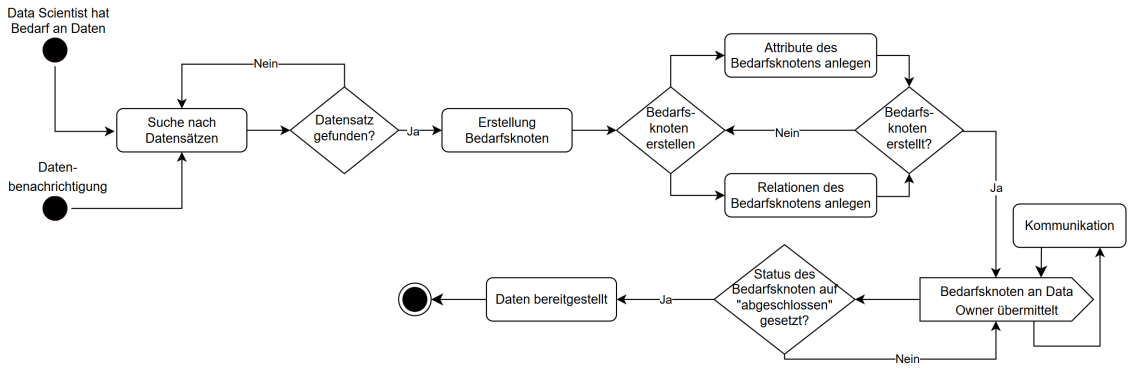


Abbildung 5.13: Ablaufdiagramm eines Data Scientists bei der Nutzung eines Wissensgraphen zur Datenabfrage

Datensatz gefunden hat, erstellt der Data Scientist einen Bedarfsknoten im Wissensgraphen. Dieser Bedarfsknoten ist ein normaler Knoten im Graphen und enthält verschiedene Attribute sowie Relationen zu anderen Knoten, die vom Data Scientist manuell festgelegt werden. Die Relationen werden zwischen dem Bedarfsknoten und den über die Suche gefundenen Knoten hergestellt, um den Datenbedarf zu verdeutlichen. Nach der Erstellung des Bedarfsknotens wird dieser an den Data Owner übermittelt, damit dieser die Daten bereitstellen kann. Der Data Owner kann den Data Scientist bei Bedarf konsultieren, sofern Fragen zur Freigabe, zu Zugriffsrechten oder zur Bereitstellung der Daten entstehen. Sobald alle Data Owner die erforderlichen Daten bereitgestellt haben, gilt der Bedarfsknoten als *abgeschlossen*. Der Data Scientist kann nun auf die bereitgestellten Daten zugreifen und diese für seine Analysen verwenden.

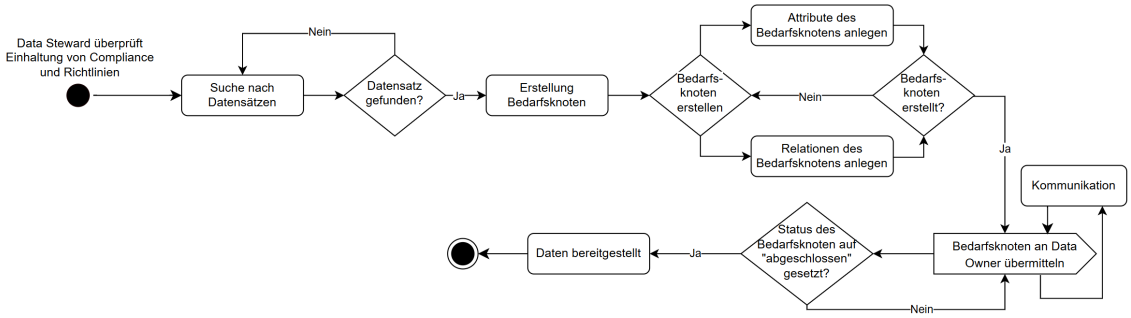


Abbildung 5.14: Ablaufdiagramm eines Data Stewards bei der Nutzung eines Wissensgraphen zur Sicherstellung der Einhaltung von Compliance-Anforderungen

Abbildung 5.14 illustriert das Ablaufdiagramm eines Data Stewards. Data Stewards tragen die Verantwortung dafür, dass Compliance-Anforderungen sowie interne Richtlinien eingehalten werden. Um dieser Aufgabe nachzukommen, durchforstet der Data Steward den Wissensgraphen, um die Knoten hinsichtlich ihrer Konformität zu überprüfen. Im Falle einer festgestellten Verletzung erstellt der Data Steward ebenfalls einen Bedarfsknoten, in dem die festgestellte Abweichung dokumentiert wird. Nach der Erstellung des Bedarfsknotens wird der Data Owner darüber informiert, um die Bereitstellung der entsprechenden Daten zu veranlassen. Der Data Owner kann den Data Steward

bei Bedarf konsultieren, sofern Fragen zur Bereitstellung der Daten entstehen. Sobald der Data Owner die notwendigen Daten aktualisiert hat, kann der Bedarfsknoten als „abgeschlossen“ markiert werden.

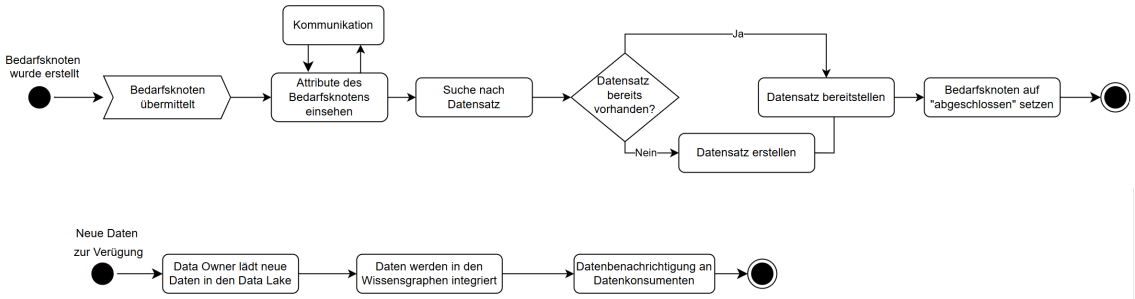


Abbildung 5.15: Ablaufdiagramm eines Data Owners bei der Nutzung eines Wissensgraphen zur Datenbereitstellung

Das Ablaufdiagramm des Data Owners in Abbildung 5.15 verdeutlicht zwei unterschiedliche Abläufe, die ein Data Owner durchlaufen kann. Das obere Diagramm stellt dabei das Pendant zu den zuvor dargestellten Ablaufdiagrammen dar. Der Data Owner wird benachrichtigt, dass ein Bedarf besteht, der von ihm erfüllt werden muss. Bei Rückfragen kann der Ersteller des Bedarfsknotens kontaktiert werden, um offene Fragen zu Berechtigungen, Freigaben und der Bereitstellung der Daten zu klären. In den bereitgestellten Informationen des Bedarfsknotens kann er die spezifische Art des Bedarfs einsehen. Anschließend hat er die Möglichkeit, nach den erforderlichen Datensätzen zu suchen. Wenn bereits ein Datensatz existiert, der die definierten Anforderungen erfüllt, können diese Daten bereitgestellt werden. Andernfalls ist der Data Owner aufgefordert, einen neuen Datensatz zu erstellen, der den Kriterien des Bedarfsknotens gerecht wird. Bei Fragen oder Unklarheiten kann er sich mit dem Ersteller des Bedarfsknotens austauschen, bevor er die Daten im Bedarfsknoten bereitstellt und den Status des Bedarfsknotens auf *abgeschlossen* setzt.

Der untere Bereich des Diagramms veranschaulicht den Ablauf, der erfolgt, wenn ein neuer Datensatz in den Data Lake und somit in den Wissensgraph integriert wird. Zunächst werden die Daten aus der entsprechenden Datenquelle in den Data Lake geladen. Im nächsten Schritt erfolgt die Abbildung dieser Daten in dem Wissensgraphen über das semantische Modell, welches in Kapitel 5.1.2 beschrieben wurde. Nachdem die Daten erfolgreich in den Wissensgraphen aufgenommen wurden, werden die Datenkonsumenten über die Verfügbarkeit der neuen Daten informiert.

6 Implementierung

In diesem Kapitel wird das im vorherigem Kapitel beschriebenen Konzept prototypisch implementiert. Da nicht das gesamte Konzept umgesetzt werden kann, liegt der Fokus darauf, die Hauptmerkmale in einem Prototypen umzusetzen. Die Hauptmerkmale beziehen sich auf die in Kapitel 5 ausgewählten Anforderungen.

6.1 Technologien

Der Protoyp besteht aus einer Webanwendung, über die ein Wissensgraph erstellt werden kann. Zur prototypischen Umsetzung des Konzepts wurde eine Kombination aus den folgenden Technologien verwendet: Als Graphdatenbank wurde die kostenfreie AuraDB [Neo24a] von Neo4j verwendet. AuraDB ist eine cloudbasierte Datenbanklösung, welche die Speicherung und Abfrage komplexer, vernetzer Datenstrukturen mithilfe der Cypher-Abfragesprache [neo24a] ermöglicht. Zur Visualisierung des Graphen wurde die Open-Source-JavaScript-Bibliothek vis-network [vis24] verwendet. Die Benutzeroberfläche wurde mithilfe der JavaScript-Bibliothek React.js [Rea24] implementiert. Um eine direkte Interaktion zwischen der Frontend-Anwendung und der Neo4j-Datenbank zu ermöglichen, wurde der Neo4j JavaScript Driver [Neo24b] genutzt. Dieser ermöglicht eine nahtlose Kommunikation zwischen der React-Anwendung und der AuraDB, sodass Daten in Echtzeit abgerufen, verarbeitet und angezeigt werden können. Node.js [Nod24] diente als Laufzeitumgebung, um die Installation von React und dessen Abhängigkeiten über den Node Package Manager (npm) zu ermöglichen. Mithilfe des Befehls *npm start* wird ein Entwicklungsserver gestartet, der die Webanwendung lokal verfügbar macht. Die Programmierung und Entwicklungsarbeit erfolgte in der integrierten Entwicklungsumgebung Visual Studio Code [Cod24], Durch die Kombination dieser Technologien konnte eine vollständige Webanwendung realisiert werden, die eine reibungslose Interaktion zwischen der Benutzeroberfläche und der Graphdatenbank ermöglicht.

Technologie	**Verwendung**
AuraDB von neo4j	Graphdatenbank
React.js	Benutzeroberflächen
Neo4j JavaScript Driver	Interaktion zwischen Frontend-Anwendung und Neo4j-Datenbank
Node.js	Laufzeitumgebung
Visual Studio Code	Entwicklungsumgebung
Vis-network	Graphvisualisierung

Tabelle 6.1: Verwendete Technologien im Prototypen

6.2 Datensatz

Im Rahmen dieser Arbeit wurde der von Neo4j bereitgestellte Datensatz Northwind [Neo24c] verwendet. Dieser Datensatz modelliert ein Produktverkaufssystem auf relationale Weise und umfasst verschiedene Domänen wie Produkte, Bestellungen, Bestelldetails, Kategorien, Lieferanten und Mitarbeiter [Neo24d]. Abbildung 6.1 zeigt ein Entity-Relationship-Diagramm (ERD) des Northwind-Datensatzes, das die strukturelle Organisation der Daten visualisiert.

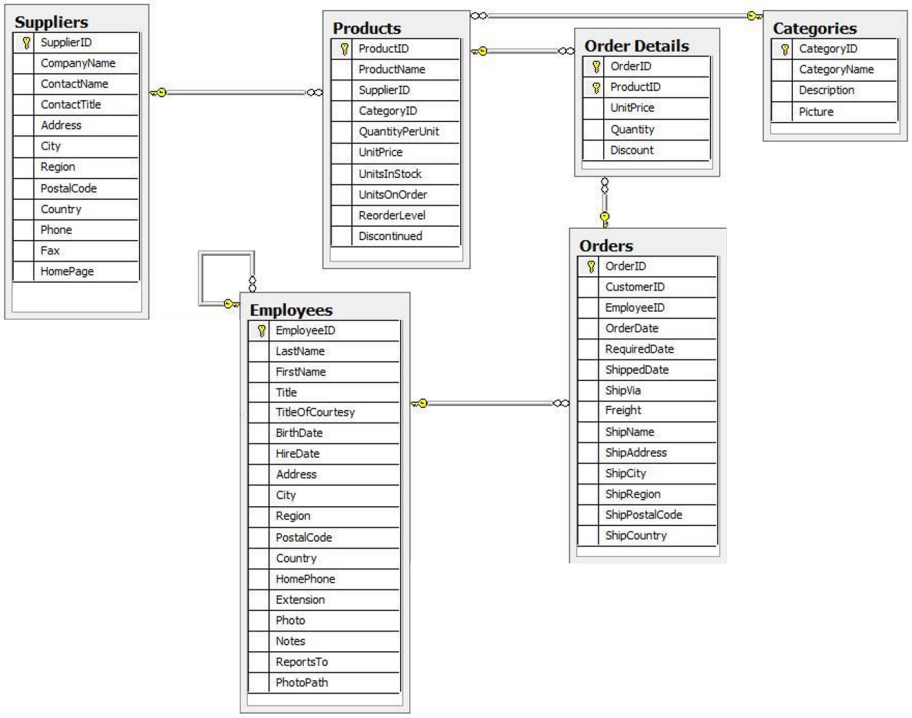


Abbildung 6.1: Entity-Relationship-Diagramm des Northwind-Datensatzes [Neo24d].

6.2.1 Domänenebene

Wie in Kapitel 5.1.1 beschrieben, wird zunächst ein Domänenmodell erstellt. Dieses ist in Abbildung 6.2 dargestellt und umfasst fünf verschiedene Knotentypen: *Mitarbeiter*, *Bestellungen*, *Produkt*, *Kategorie* und *Anbieter*. Diese Knotentypen entsprechen den Tabellen, die in Abbildung 6.1 abgebildet sind. Zunächst werden diese als Knoten mit dem Label *Oberkategorie* in den Graphen integriert. Die Labels sind für das spätere Kapitel wie 6.3.5 wichtig. Anschließend werden, basierend auf den Beziehungen aus Abbildung 6.2, Verknüpfungen zwischen den Knoten hinzugefügt, bis das Domänenmodell vollständig aufgebaut ist. In dem Listing 6.1 wird beispielhaft die Erstellung der Knoten *Produkt* und *Bestellung* gezeigt, sowie die Erstellung der Beziehungen *ENTHÄLT* zwischen den beiden.

Das entwickelte Domänenmodell weist eine generische und oberflächliche Struktur auf, die für die Zielsetzung des Wissensgraphen in dieser Arbeit nur bedingt geeignet ist. Ziel des Wissensgraphen ist es, die Vermittlung von Daten zwischen Datenproduzenten und -konsumenten

zu optimieren. Insbesondere soll der Wissensgraph Datenkonsumenten, wie beispielsweise Data Scientists, dabei unterstützen, relevante Daten für ihre Analysen effizient zu identifizieren. Ein konkretes Anwendungsbeispiel ist die Durchführung einer Analyse zu einem bestimmten Produkt. Für solche Szenarien reicht das vorhandene Domänenmodell nicht aus, da es nicht genügend Detailtiefe aufweist, um einen entsprechenden Bedarfsknoten abzubilden. Um dies zu ermöglichen, müssen demnach auch einzelne Instanzen des Domänenmodells als individuelle Knoten in den Graphen integriert werden.

```
// Erstellung der Knoten
CREATE (p:Oberkategorie {name: 'Produkt'})
CREATE (p:Oberkategorie {name: 'Bestellung'})

// Erstellung Beziehung
MATCH (p:Oberkategorie {name: 'Produkt'}), (b:Oberkategorie {name: 'Bestellung'})
CREATE (b)-[:ENTHÄLT]->(p)
```

Listing 6.1: Erstellung der Knoten und Beziehungen des Domänenmodells.

Um entscheiden zu können, welche Instanzen benötigt werden, ist die systematische Unterteilung der Daten in Stamm- und Massendaten, wie in Kapitel 5.1.1 beschrieben, notwendig. Stammdaten sind relativ stabil und ändern sich nur selten, während Massendaten dynamisch und kurzlebig sind. Im vorliegenden Datensatz gehören die Informationen in der Tabelle *Bestellung* zu den Massendaten, da sie in großen Mengen generiert werden und eine Vielzahl an Informationen wie Transaktionen, Produkte, Kunden und Lieferungen umfassen. Im Gegensatz dazu gelten die Daten zu Produkten, Mitarbeitern, Kategorien und Anbietern als Stammdaten, da sie grundlegende und wiederverwendbare Informationen darstellen, die über längere Zeiträume hinweg stabil bleiben.

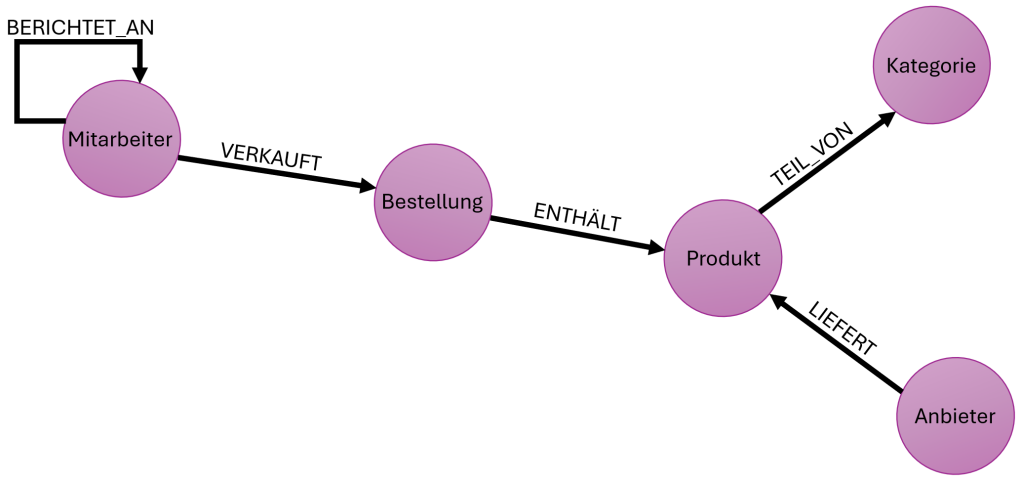


Abbildung 6.2: Domänenmodell des Northwind-Datensatzes in Anlehnung an [Neo24d].

Im Kontext der Oberkategorie *Bestellung* werden keine weiteren Details in den Wissensgraphen integriert. Im Gegensatz dazu wird aus den Tabellen der anderen vier Oberkategorien (*Produkt*, *Mitarbeiter*, *Kategorie*, *Anbieter*) jede einzelne Zeile als Knoten in den Graphen überführt.

Wie in Kapitel 5.1.2 erläutert, ist ein semantisches Modell erforderlich, um Daten aus dem Data Lake in die Struktur eines Wissensgraphen zu integrieren. Dieses Modell dient als Abbildung zwischen den Entitäten und den Beziehungen des Wissensgraphen und den zugrunde liegenden Datenquellen.

Für den betrachteten Datensatz wird das semantische Modell wie folgt definiert: Zeilen der Tabellen werden zu Knoten im Wissensgraphen, Tabellennamen fungieren als Beschriftungen der Knoten, Fremdschlüssel oder definierte Joins zwischen den Tabellen bilden die Beziehungen zwischen den Knoten.

Beispielsweise wird jede Zeile der Produkt-Tabelle als ein Knoten mit der Beschriftung *Produkt* in den Graphen aufgenommen. Dabei werden jedoch nur ausgewählte Attribute übernommen: Der Name des Knotens sowie die Attribute, die Beziehungen repräsentieren, wie etwa Fremdschlüssel. Alle anderen Attribute bleiben unberücksichtigt, da der Wissensgraph in diesem Fall ausschließlich als Datenkatalog fungiert. Der Fokus liegt darauf, einen Verweis auf die bestehenden Daten zu schaffen, anstatt alle Datensätze vollständig im Graphen abzubilden.

Das Listing 6.2 zeigt, wie Produktknoten erstellt werden. Dabei wird der Name des Produkts als Knotenattribut gespeichert. Zusätzlich werden die Fremdschlüssel integriert, die die Beziehungen zur Kategorie und zum Anbieter des Produkts darstellen. Da sich die Daten in einer CSV-Datei in einem Github-Repository [Neo24c] befinden, müssen die Daten hier ausgelesen werden. Ein Ausschnitt des Codes verdeutlicht die Erstellung der Produktknoten. Die übrigen Knoten der Tabellen *Mitarbeiter*, *Kategorie* und *Anbieter* werden nach dem selben Prinzip integriert.

```
LOAD CSV WITH HEADERS FROM 'https://gist.githubusercontent.com/jexp/054bc6baf36604061
    bf407aa8cd08608/raw/8bdd36dfc88381995e6823ff3f419b5a0cb8ac4f/products.csv' AS row
MERGE (produkt:Produkt {productID: row.ProductID})
ON CREATE SET
produkt.productName = row.ProductName,
produkt.supplierID = row.SupplierID,
produkt.categoryID = row.CategoryID;
```

Listing 6.2: Erstellung der Produktknoten.

Nachdem alle Knoten erstellt worden sind, folgt die Modellierung der Kanten. Diese Verbindungen repräsentieren die Beziehungen zwischen den verschiedenen Elementen im Wissensgraphen und tragen spezifische Bezeichnungen, die den Zusammenhang der Knoten zueinander verdeutlichen. Ein konkretes Beispiel hierfür ist die Beziehung zwischen den Produkten und den Kategorien. In diesem Fall wird eine Verbindung mit der Bezeichnung *TEIL_VON* angelegt, die ausdrückt, dass ein Produkt einer bestimmten Kategorie zugeordnet ist. Der im Listing 6.3 aufgeführte Code demonstriert die Implementierung dieser Beziehung. Nachdem alle Beziehungen gemäß Abbildung 6.2 angelegt wurden, ist die Domänenebene aus dem Konzept vollständig erstellt.

```
LOAD CSV WITH HEADERS FROM 'https://gist.githubusercontent.com/jexp/054bc6baf36604061
bf407aa8cd08608/raw/8bdd36dfc88381995e6823ff3f419b5a0cb8ac4f/products.csv' AS row
MATCH (produkt:Produkt {productID: row.ProductID})
MATCH (kategorie:Kategorie {categoryID: row.CategoryID})
MERGE (produkt)-[:TEIL_VON]->(kategorie);
```

Listing 6.3: Erstellung Beziehungen zwischen Produkten und Kategorien.

Zum Schluss muss noch eine Verbindung zwischen den einzelnen Knoten der Instanzen und dem passenden Knoten des Domänenmodells erstellt werden. Hierfür wird, wie in Listing 6.4 dargestellt, eine *INSTANZ_VON*-Beziehung zwischen den Produktinstanzen und dem Produktknoten im Domänenmodell erstellt. Dasselbe wird für die anderen Instanzen der Domänenebene ebenfalls durchgeführt.

```
MATCH (o:Oberkategorie {name:'Produkt'}), (p:Produkt)
MERGE (p)-[:INSTANZ_VON]->(o)
```

Listing 6.4: Erstellung Beziehung zwischen Instanzen und Domänenmodell.

Im Anhang A1 ist die erstellte Domänenebene im Neo4j-Browser [neo24b] dargestellt. Die Knoten des Domänenmodells sind hier blau hinterlegt. In der Abbildung wird deutlich, dass der Knoten *Bestellung*, der sich in unten in der Mitte der Darstellung befindet, keine weiteren Instanzen besitzt. Die Instanzen von *Mitarbeiter* sind grün dargestellt, die Produktinstanzen in Pink, die Kategorieinstanzen in Lila, und die Anbieter sind in einem hellen Braun hervorgehoben.

6.2.2 Datensatzreferenzebene

Die Datensatzreferenzebene ist, wie in Kapitel 5.1.1 beschrieben, die Schicht des Wissensgraphen, die Verweise auf die Speicherorte der Datensätze enthält. Da der Wissensgraph primär als Datenkatalog dient, werden die Datensätze selbst nicht direkt in den Wissensgraphen integriert. Stattdessen werden lediglich Referenzknoten erstellt, die auf den Speicherort der Datensätze im Data Lake referenzieren. In Listing 6.5 ist beispielhaft die Erstellung des Datensatzreferenzknotens *Bestellungtabelle* zu sehen. Für jeden der fünf Domänenknoten wird ein Datensatzreferenzknoten erstellt. Dieser verweist auf eine der fünf CSV-Dateien, in denen die Daten gespeichert sind. Diese Dateien wurden zur Erstellung der Domänenebene verwendet, wie in Listing 6.2 gezeigt.

```
CREATE (:Datensatzreferenz {
    Name: 'Bestellungtabelle',
    Verlinkung: 'https://gist.githubusercontent.com/jexp/054bc6baf36604061f407aa8cd08608
    /raw/8bdd36dfc88381995e6823ff3f419b5a0cb8ac4f/order.csv',
    Beschreibung: 'Diese Tabelle enthält alle Daten zu den Bestellungen'
})
```

Listing 6.5: Erstellung eines Datensatzreferenzknoten.

Die Cypher-Anfrage in Listing 6.5 erstellt einen Knoten mit dem Namen *Bestellungtabelle* und weist ihm das Label *Datensatzreferenz* zu. Der Knoten erhält zwei Attribute: Zum einen den Link, welcher auf die CSV-Datei verweist, die die Daten zu den Bestellungen enthält. Zum anderen enthält er eine Beschreibung, in welcher eine Erläuterung der enthaltenen Daten steht. Nach der Erstellung des Knotens wird eine Beziehung zwischen der neu erstellten *Bestellungtabelle* und dem Knoten *Bestellung* aus der Domänenebene hergestellt, was in Listing 6.6 zu sehen ist. Diese Beziehung verbindet die Datensatzreferenzebene mit der Domänenebene.

```
MATCH (oberkategorie:Oberkategorie {name: 'Bestellung'}),
  (referenz:Datensatzreferenz {Name: 'Bestellungtabelle'})
MERGE (oberkategorie)-[:ZU_FINDEN_IN]->(referenz);
```

Listing 6.6: Beziehung zwischen Domänenebene und Datensatzreferenzebene.

Für jeden Knoten mit dem Label *Oberkategorie* kann ein entsprechender Knoten in der Datensatzreferenzebene erstellt werden, der auf die Tabelle verweist, aus der dieser stammt. Dabei werden Datensatzreferenzknoten ausschließlich für die Domänenknoten angelegt, nicht jedoch für einzelne Instanzen. Dies geschieht aus verschiedene Gründen. Erstens dient es der Übersichtlichkeit: Wenn für jede Instanz ein Knoten in der Datensatzreferenzebene erstellt wird, wird der Wissensgraph groß

und unübersichtlich. Zweitens kann der Speicherplatz einer einzelnen Instanz auch transitiv über den Knoten des Domänenmodells ermittelt werden. Dieser verweist auf die dazugehörige Tabelle, und die ID der Instanz in der Tabelle identifiziert den passenden Datensatz.

6.2.3 Datenproduktebene

Wie in Abbildung 5.1 abgebildet, liegt unterhalb der Domänenebene die Datenproduktebene. Datenprodukte definieren den Rahmen, wie die zugrundeliegenden Datensätze genutzt werden. Sie strukturieren die Daten und legen fest,wie diese gesammelt, verarbeitet und analysiert werden, um für den jeweiligen Anwendungsfall relevante Einblicke oder Funktionen bereitzustellen. Da in dem Northwind-Datensatz keine Datenprodukte enthalten sind, wurden für diesen Prototypen beispielhaft eigene Datenprodukte erstellt, um das Konzept zu veranschaulichen.

Im dem Listing 6.7 wird ein Datenprodukt erstellt, welches einen Report über den Verkauf von Produkten pro Mitarbeiter enthält. Es wird ein Knoten mit dem Label *Datenprodukt* erstellt, welcher den Namen und eine Beschreibung als Attributwerte übergeben bekommt, um nachvollziehen zu können, was in diesem Datenprodukt enthalten ist. Im Anschluss daran wird der Datenproduktknoten mit den Knoten der Domänenebene verbunden, welche für diese Auswertung benötigt wurden. Dazu gehören die Knoten *Bestellung*, *Produkt* und *Mitarbeiter*. In dem folgenden Beispiel wurde der Datenproduktknoten ausschließlich mit den Oberkategorien der Domänenebene verknüpft, da alle Daten benötigt werden und nicht nur einzelne Produkte oder Mitarbeiter.

```
CREATE (dp:Datenprodukt {
Name: 'AuswertungVerkaufProMitarbeiter',
Beschreibung: 'Dieses Datenprodukt enthält einen Report über den Verkauf von Produkten
              pro Mitarbeiter'
})
WITH dp
MATCH (ok:Oberkategorie {name: 'Bestellung'}),
       (ok2:Oberkategorie {name: 'Produkt'}),
       (ok3:Oberkategorie {name: 'Mitarbeiter'}),
      (dsr:Datensatzreferenz {Name: 'Bestellungtabelle'}),
      (dsr2:Datensatzreferenz {Name: 'Produkttabelle'}),
      (dsr3:Datensatzreferenz {Name: 'Mitarbeitertabelle'})
CREATE (dp)-[:VERWENDET]->(ok),
       (dp)-[:VERWENDET]->(ok2),
       (dp)-[:VERWENDET]->(ok3),
       (dp)-[:NUTZT]->(dsr),
       (dp)-[:NUTZT]->(dsr2),
       (dp)-[:NUTZT]->(dsr3);
```

Listing 6.7: Erstellung des Datenproduktes *AuswertungVerkaufProMitarbeiter*.

Das nächste Listing 6.8 illustriert, wie ein Datenprodukt erstellt wird, das mit bestimmten Produkten verknüpft ist. Im gezeigten Beispiel repräsentiert der Datenproduktknoten die Auswertung eines spezifischen Produkts – in diesem Fall *Tofu*. Es gibt verschiedene Arten von Beziehungen bei diesem Datenprodukt: Zum einen eine Verbindung zur Oberkategorie *Bestellung*. Diese zeigt auf eine allgemeine Kategorie, da keine spezifischen Auswertungen für einzelne Bestellungen vorliegen. Dann gibt es eine feingranulare Verbindung, welche zu einem Knoten auf der Domänenebene führt,

der das Produkt *Tofu* repräsentiert. Zudem gibt es eine Verbindung zu einem Datenproduktknoten. Dieser Knoten enthält bereits Auswertungen für alle Produkte, aus denen benötigten Informationen für das Produkt *Tofu* extrahiert werden können.

```
    CREATE (dp:Datenprodukt {
    Name: 'AuswertungTofu',
    Beschreibung: 'Dieses Datenprodukt enthält einen Report über den Verkauf von dem Produkt
Tofu'
    })
    WITH dp
    MATCH (p:Produkt {productName: 'Tofu'}),
          (ok:Oberkategorie {name: 'Bestellung'}),
          (dsr:Datensatzreferenz {Name: 'Bestellungtabelle'}),
          (dsr2:Datensatzreferenz {Name: 'Produkttabelle'}),
          (dp2:Datenprodukt {Name: 'AuswertungProdukte'})
    CREATE (dp)-[:VERWENDET]->(ok),
           (dp)-[:VERWENDET]->(p),
           (dp)-[:VERWENDET]->(dp2),
           (dp)-[:NUTZT]->(dsr),
           (dp)-[:NUTZT]->(dsr2);
```

Listing 6.8: Erstellung des Datenproduktes *AuswertungTofu*.

6.2.4 Metadatenebene

Wie in Abbildung 5.1 zu sehen ist, zieht sich die Metadatenebene durch alle vorher beschriebenen Ebenen, weshalb für jede Metadatenart ein eigenes Label erstellt werden muss. Der Grund wird in Kapitel 6.3.5 genauer erörtert. Zudem besitzen die verschiedenen Metadatenarten unterschiedliche Attribute basierend auf ihren Eigenschaften, welche in Kapitel 3.1 erörtert wurden. Somit besitzt jede Art von Metadaten-Knoten eine unterschiedliche Ontologie. Im Folgenden werden alle drei Knotenarten vorgestellt. Da in dem Northwind-Datensatz jedoch keine Metadaten enthalten sind, wurden die Inhalte der Attribute hier improvisiert, um das Konzept und die Idee zu veranschaulichen.

```
    CREATE (om:OperativeMetadaten {
        Name: 'OperativeMetadatenAuswertungTofu',
        Beschreibung: 'Operative Metadaten zu dem Datenprodukt AuswertungTofu',
        DataLineage: 'Rohdaten aus Produkttabelle Reihe 10 für Daten zum Produkt Tofu;
                        Rohdaten aus Bestellungstabelle Reihe 13 - 37'
    })
    WITH om
    MATCH (p:Produkt {productName: 'Tofu'}),
          (b:Oberkategorie {name: 'Bestellung'}),
          (d:Datensatzreferenz  {Name: 'Reports'})
    CREATE (om)-[:BEINHALTET]->(p),
           (om)-[:BEINHALTET]->(b),
           (om)-[:ZU_FINDEN_IN]->(d);
```

Listing 6.9: Erstellung operativer Metadaten.

Zu Beginn werden die operativen Metadaten erstellt, wie in Listing 6.9 zu sehen ist, welche zentrale Aspekte der Datenverarbeitung umfassen, wie beispielsweise die Data Lineage. Aus diesem Grund tragen die Knoten der operativen Metadaten das Attribut *DataLineage*, welches auf der

Datensatzreferenzebene benötigt wird, um die Herkunft und Verarbeitung der Daten nachverfolgen zu können. Im folgenden Listing 6.9 wird ein Knoten mit dem Label *OperativeMetadaten* erstellt. Dieser Knoten repräsentiert die operativen Metadaten des Produkts *AuswertungTofu*. Zum einen wird hier die Verbindung zu dem Produkt *Tofu* und den Bestellungen hergestellt, da diese Auswertung basierend auf diesen Daten stattgefunden hat. Zum anderen wird noch eine Relation hinzugefügt, die eine Verbindung zur Datensatzreferenzebene herstellt, in diesem Fall zu dem Knoten *Reports*, in welcher die Daten zu der erstellten Auswertung referenziert werden.

Im nächsten Schritt werden die technischen Metadaten erstellt, wie in Listing 6.10 zu sehen ist. Diese enthalten Informationen über das Format, die Struktur des Datensatzes, Datentypen, Datengröße und Zugriffsrechte. Diese Informationen werden als Attribute in den Knoten aufgenommen, um die technischen Eigenschaften des Datensatzes zu dokumentieren. Der folgende Code erstellt einen Knoten, der die technischen Metadaten des Knotens *Produkttabelle* repräsentiert.

```
    CREATE (tm:TechnischeMetadaten {
        Name: 'TechnischeMetadatenProduktttabelle',
        Beschreibung: 'Technische Metadaten zu der Produkttabelle',
        Format: 'CSV-Datei',
        StrukturDatensatz: 'Spalten: ProductID, ProductName, SupplierID, CategoryID,
QuantityPerUnit, UnitPrice, UnitsInStock, UnitsOnOrder, RecorderLevel, Discontinued',
        Datentypen: 'Integer, String, Integer, Integer, String, Float, Integer, Integer,
Integer, Boolean',
        Datengröße: '4.23 KB',
        Zugriffsrechte: 'Personen der Berechtigungsgruppe TM'
    })
    WITH tm
    MATCH (d:Datensatzreferenz {Name: 'Produkttabelle'})
    CREATE (tm)-[:GEHÖRT_ZU]->(d);
```

Listing 6.10: Erstellung technischer Metadaten.

Abschließend werden in Listing 6.11 die geschäftlichen Metadaten erstellt. Diese Metadaten dienen dazu, den Daten einen Kontext und eine Bedeutung zu verleihen, sodass Datenkonsumenten die Daten besser verstehen und interpretieren können. In diesem Beispiel wurde das Attribut Kontext hinzugefügt, um den Daten eine erklärende Dimension zu geben. Es können jedoch auch weitere Attribute aufgenommen werden, je nach den spezifischen Anforderungen und der Art der Daten, um diesen noch mehr Bedeutung zu verleihen und eine präzisere Interpretation zu ermöglichen.

```
    CREATE (gm:GeschäftlicheMetadaten {
        Name: 'GeschäftlicheMetadatenTofu',
        Beschreibung: 'Geschäftliche Metadaten zu Produkt Tofu',
        Kontext: 'Artikelnummer: 12345-TOFU; Produktkategorie: Lebensmittel - Vegetarische/
            vegane Produkte; Preis: 2,99 Euro pro 250g-Packung, Produktionskosten: 1,20 Euro
            pro Einheit; Beliebte Absatzmärkte: Asien, Europa, Nordamerika;
            Hauptvertriebskanäle: Supermärkte, Bioläden, Online-Handel; Zielgruppen:
            Vegetarier, Veganer, gesundheitsbewusste Konsumenten '
    })
    WITH gm
    MATCH (p:Produkt {productName: 'Tofu'})
    CREATE (gm)-[:GEHÖRT_ZU]->(p);
```

Listing 6.11: Erstellung geschäftlicher Metadaten.

6.3 Benutzeroberfläche

In diesem Kapitel werden die einzelnen Komponenten der entwickelten Benutzeroberfläche erläutert. Dabei wird beschrieben, wie die Oberfläche aufgebaut ist, wie diese funktioniert und wie sie das zuvor dargestellten Konzept umsetzt.

6.3.1 Erstellung von Bedarfsknoten

Der Fokus der Implementierung liegt auf der Erstellung eines Bedarfsknoten auf der Datenproduktebene, um den Bedarf an Daten für beispielsweise eine Analyse zu vermerken. In dem Abschnitt *Bedarfsknoten erstellen* befindet sich ein Formular, welches in Abbildung 6.3 zu sehen ist. Das Formular dient der Erstellung eines Bedarfsknotens und basiert auf der in Abbildung 5.4 dargestellten Ontologie.

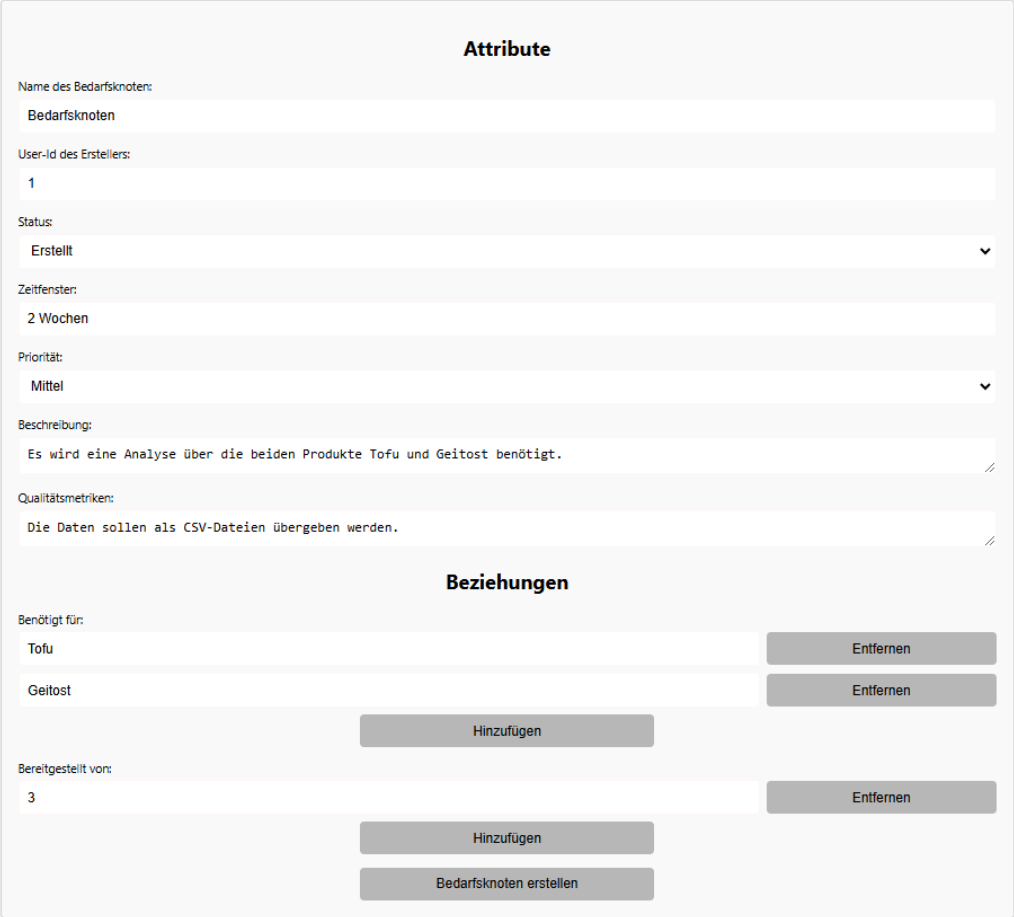


Abbildung 6.3: Formular zur Erstellung eines Bedarfsknotens im User Interface.

Zunächst müssen alle Attribute ausgefüllt werden, die in Kapitel 5.3.1 ausführlich beschrieben sind. Für die meisten Felder ist Freitext vorgesehen, während für den Status und die Priorität ein Dropdown-Menü mit den verschiedenen Optionen bereitgestellt wird. Die eingegebenen Werte werden anschließend in einem String gespeichert. Darauf folgen die Beziehungen, welche ebenfalls

die in Kapitel 5.3.1 erläutert wurden. Auch hier ist Freitext vorgesehen, wobei in jedes Feld nur ein Produkt oder eine ID eingegeben werden darf. Um bei Bedarf weitere Beziehungen hinzuzufügen, wurde ein Button erstellt, der ein weiteres Beziehungfeld zur Verfügung stellt. Die Beziehungen werden in Arrays gespeichert.

Nachdem alle Eingaben abgeschlossen sind, wird eine Cypher-Anfrage ausgeführt, um den Bedarfsknoten zu erstellen, welche im Listing 6.12 zu sehen ist. Die eingegebenen Werte aus dem Formular werden dabei an die Cypher-Anfrage übergeben, die daraufhin die Anfrage mit den gesetzten Werten an die Datenbank sendet und einen Bedarfsknoten anlegt. In der Cypher-Anfrage wird ein Knoten mit dem Label *Bedarf* erstellt, um ihn von anderen Knoten zu unterscheiden. Zunächst werden alle Attribute des Bedarfsknotens deklariert, denen die Eingabewerte einfach zugewiesen werden können. Dies ist im oberen Teil der Cypher-Anfrage zu sehen.

```
CREATE (d:Bedarf {
        name: $name,
        status: $status,
        timeFrame: $timeFrame,
        priority: $priority,
        description: $description,
        qualityMetrics: $qualityMetrics,
        requiredFor: $requiredFor,
        providedBy: $providedBy,
        userID: $userID
    })
    WITH d, $requiredFor AS requiredFor, $providedBy AS providedBy, $userID AS userID

    UNWIND requiredFor AS req
    MATCH (n)
    WHERE n.productName = req
        OR n.companyName = req
        OR n.orderID  = req
        OR n.categoryName  = req
        OR n.name  = req
    CREATE (d)-[r:BENÖTIGT]->(n)
    WITH d, r, n, providedBy, userID

    UNWIND providedBy AS prov
    MERGE (e:Mitarbeiter {employeeID: prov})
    MERGE (d)-[r2:BEREITGESTELLT_VON]->(e)
    WITH d, r, n, r2, e, userID

    MERGE (e2:Mitarbeiter {employeeID: userID})
    MERGE (e2)-[r3:ERSTELLT]->(d)

    RETURN d, r, n, r2, e, r3, e2
```

Listing 6.12: Cypher-Anfrage zur Erstellung eines Bedarfsknoten.

Bei den Relationen gestaltet sich dies jedoch komplexer. Da *requiredFor* und *providedBy* als Arrays deklariert wurden, um mehrere Beziehungen zu ermöglichen, können die Eingabewerte nicht direkt zugewiesen werden. Deshalb müssen, wie im Code ersichtlich, die Arrays zunächst entpackt werden. In Cypher wird hierfür *UNWIND* verwendet, um Arrays in einzelne Elemente zu zerlegen und

für jedes Element im Array eine entsprechende Relation zu erstellen. Die Relationen *REQUIRES* und *PROVIDED_BY* verbinden den Bedarfsknoten mit den anderen Knoten, während die Relation *CREATED* eine Verbindung vom Mitarbeiter zum Bedarfsknoten herstellt.

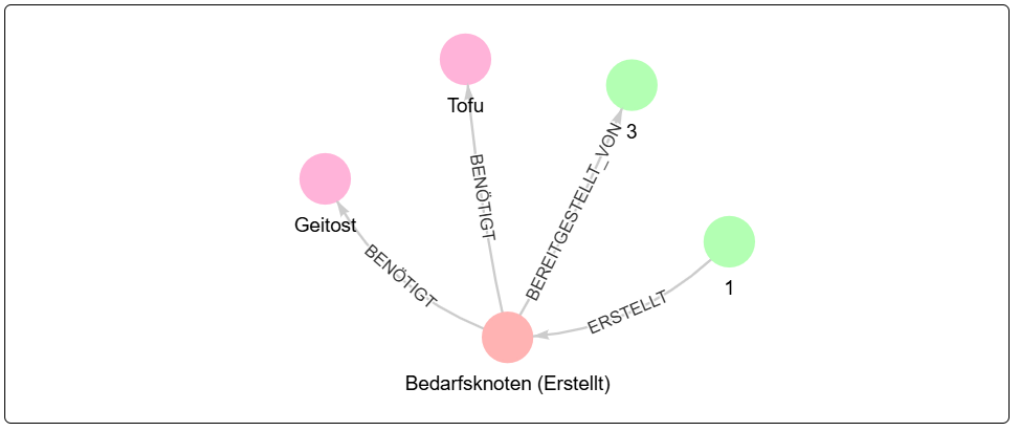


Abbildung 6.4: Formular zur Erstellung eines Bedarfsknoten.

Nachdem der Bedarfsknoten erstellt wurde, wird dieser angezeigt, wie in Abbildung 6.4 dargestellt. Dafür wurde wie bereits erwähnt die Open-Source-JavaScript-Bibliothek vis-network [vis24] verwendet. Diese Bibliothek ermöglicht die interaktive und visuelle Darstellung von Netzwerken oder Graphen. Das zentrale Element der Bibliothek ist die Klasse *Network*, die zur Erstellung und Darstellung eines Netzwerks aus Knoten und Kanten dient. Mithilfe dieser Klasse können sowohl die Darstellung des Netzwerks gesteuert als auch interaktive Funktionen wie Zoom implementiert werden. Die zugehörige Cypher-Abfrage, die den Bedarfsknoten abruft, basiert auf der Übergabe des eingegebenen Namens im Formular an eine weitere Cypher-Anfrage, die den Knoten abruft. Listing 6.13 zeigt sowohl Kanten, die vom Bedarfsknoten wegführen, als auch Kanten, die auf den Bedarfsknoten zeigen. Dadurch wird auch die *ERSTELLT*-Beziehung berücksichtigt.

```
MATCH (d:Bedarf)-[r]->(n)
WHERE d.name = $name
RETURN d, r, n
UNION
MATCH (n)-[r]->(d:Bedarf)
WHERE d.name = $name
RETURN n, r, d
```

Listing 6.13: Cypher-Anfrage um Bedarfsknoten anzuzeigen.

Die Cypher-Anfragen gibt den erstellten Bedarfsknoten zurück, also die Knoten und Kanten, welche basierend auf der Anfrage gefunden wurden. Diese werden dann mittels des vis-network graphisch aufbereitet und visualisiert, wie in Abbildung 6.4 zu sehen ist. Die Farben entsprechen denen der in der Domänenebene in Kapitel 6.2 verwendeten Farben, mit dem Zusatz, dass Bedarfsknoten in rot dargestellt werden. Neben dem Namen des Bedarfsknotens wird auch der Status in Klammern angezeigt. Die Anzeige des Status bietet eine klare Übersicht über den Fortschritt des Bedarfsknotens und ermöglicht es, diesen auf einen Blick zu erfassen. Ein weiterer wichtiger Aspekt ist, dass der Bedarfsknoten nach dem Setzen des Status auf *abgeschlossen* für

eine gewisse Zeit im Wissensgraphen verbleibt. Dies erfolgt, wie in Kapitel 5.3.4 beschrieben, aus Nachvollziehbarkeitsgründen, um die Historie des Bedarfsknotens zu bewahren und eine spätere Rückverfolgbarkeit zu gewährleisten.

6.3.2 Benutzereingabe

Nachdem ein Bedarfsknoten erstellt wurde, müssen die Personen informiert werden, die zur Erfüllung des Bedarfsknotens beitragen können, also diejenigen, die in Kapitel 6.3.1 über die *PROVIDED_BY*-Verbindung mit dem Bedarfsknoten verknüpft wurden. Dafür wurde der Abschnitt *Benutzereingabe* entwickelt, welcher in Abbildung 6.5 zu sehen ist. In diesem Abschnitt erfolgt die Eingabe der Benutzer-ID. In einer realen Unternehmensumgebung ist eine manuelle Eingabe solcher Daten in der Regel nicht erforderlich, da diese bei der Anmeldung über Benutzernamen und Passwort oder über Single-Sign-On (SSO)[1] automatisch übermittelt werden. Da es sich hierbei jedoch um einen Prototyp handelt, ist die manuelle Eingabe der User- ID erforderlich.

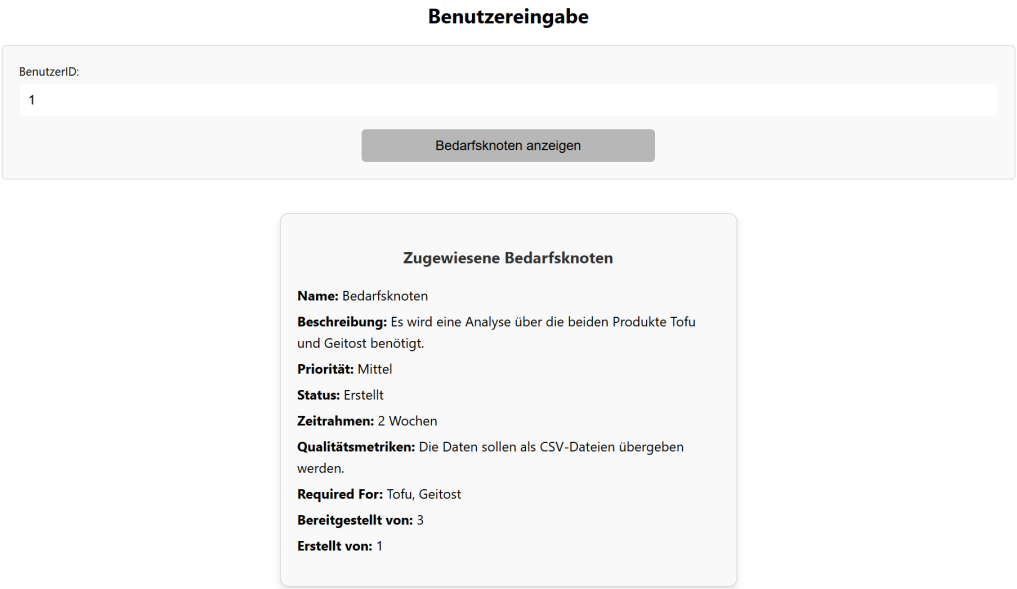


Abbildung 6.5: Benutzereingabe des User Interface.

```
MATCH (d:Bedarf)-[:BEREITGESTELLT_VON]->(e:Mitarbeiter {employeeID: $userId})
RETURN d
UNION
MATCH (e:Mitarbeiter {employeeID: $userId})-[:ERSTELLT]->(d:Bedarf)
RETURN d
```

Listing 6.14: Anzeigen zugewiesener Bedarfsknoten.

[1] Single Sign-On ist ein Authentifizierungsverfahren, das es Nutzern ermöglicht, mit einer einzigen Anmeldung Zugriff auf mehrere Anwendungen oder Systeme zu erhalten, ohne sich erneut authentifizieren zu müssen. [Sca24]

Nachdem die Nutzer-ID eingegeben wurde, werden, wie in Abbildung 6.5 dargestellt, die Bedarfsknoten angezeigt, die der jeweiligen Person zugeordnet sind. Dies umfasst sowohl die Bedarfsknoten, die die Person selbst erstellt hat, als auch jene, die ihr zugewiesen wurden. Zu diesem Zweck wurde die Cypher-Abfrage in Listing 6.14 formuliert, die die eingegebene ID verwendet, um die Bedarfsknoten zu filtern, die entweder dem Nutzer zur Bereitstellung von Daten zugewiesen wurden oder die er selbst erstellt hat.

6.3.3 Änderung des Status des Bedarfsknoten

Nachdem ein Benutzer die Daten bereitgestellt hat, ist es wichtig, den Ersteller des Bedarfsknotens über den Fortschritt zu informieren. Zu diesem Zweck wurde der Abschnitt *Bedarfsknoten Status ändern* entwickelt, der in Abbildung 6.6 gezeigt wird. Nach der Eingabe des Namens des Bedarfsknotens und der Statusänderung wird die Cypher-Abfrage aus Listing 6.15 ausgeführt:

```
MATCH (d:Bedarf {name: $demandName})
SET d.status = $status
RETURN d;
```

Listing 6.15: Cypher-Anfrage um den Status des Bedarfsknotens zu ändern.

Somit wird der Status des Bedarfsknotens aktualisiert. Auf diese Weise werden alle Beteiligten, sowohl der Ersteller des Bedarfsknotens als auch die anderen Datenbereitsteller, über den aktuellen Stand informiert.

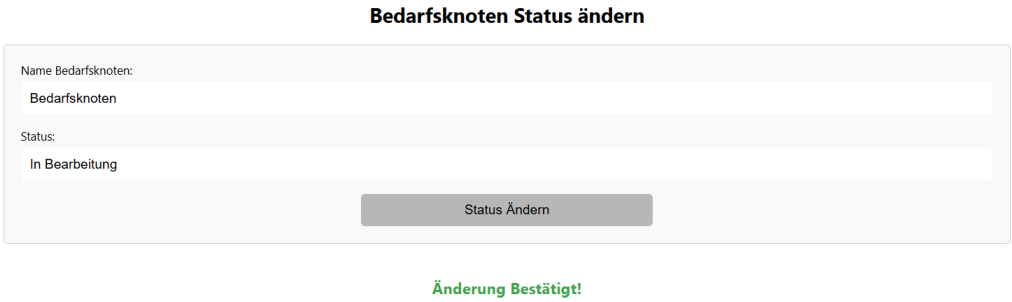


Abbildung 6.6: Änderung des Status des Bedarfsknoten über User Interface.

6.3.4 Suche im Wissensgraph

Ein Vorteil bei der Verwendung von Wissensgraphen besteht darin, dass eine umfassende Suche ermöglicht wird und dabei Verbindungen zwischen Daten sichtbar gemacht werden, die zuvor nicht erkennbar waren. Aus diesem Grund wird eine Suchfunktion innerhalb des Wissensgraphen bereitgestellt. Sie hilft dabei, neue und potenziell interessante Datensätze zu finden. Außerdem können damit Datensätze identifiziert werden, die mit dem aktuell benötigten Datensatz in Verbindung stehen und ebenfalls relevant sein könnten. Zudem muss der Benutzer bei der Erstellung des Bedarfsknotens, wie in Kapitel 6.3.1 beschrieben, die Namen der Knoten angeben, die im Graphen zu finden sind. Nach Eingabe des Suchbegriffs und der Rolle, die der Nutzer im Unternehmen innehat, wird der Graph basierend auf dem Suchbegriff durchforstet.

```
MATCH (n)-[r]->(m)
    WHERE n.productName =~ '(?i).*' + $search + '.*'
    OR n.name  =~ '(?i).*' + $search + '.*'
    OR n.Name  =~ '(?i).*' + $search + '.*'
    OR n.companyName =~ '(?i).*' + $search + '.*'
    OR n.employeeID  =~ '(?i).*' + $search + '.*'
    OR n.categoryName  =~ '(?i).*' + $search + '.*'
    RETURN n, r, m
```

Listing 6.16: Suche im Wissensgraph.

Die in Listing 6.16 formulierte Anfrage sucht nach Knoten, die den Suchbegriff enthalten und zeigt diese, sowie die über eine Relation verbundenen Knoten visuell an, wie in Abbildung 6.7 zu sehen ist. Auch hier wurden die Farben der Knoten basierend auf den in Kapitel 6.2 beschriebenen Farben gewählt. Dabei ist es unerheblich, an welcher Stelle im Knoten der Begriff vorkommt oder ob davor oder danach noch andere Zeichen stehen.

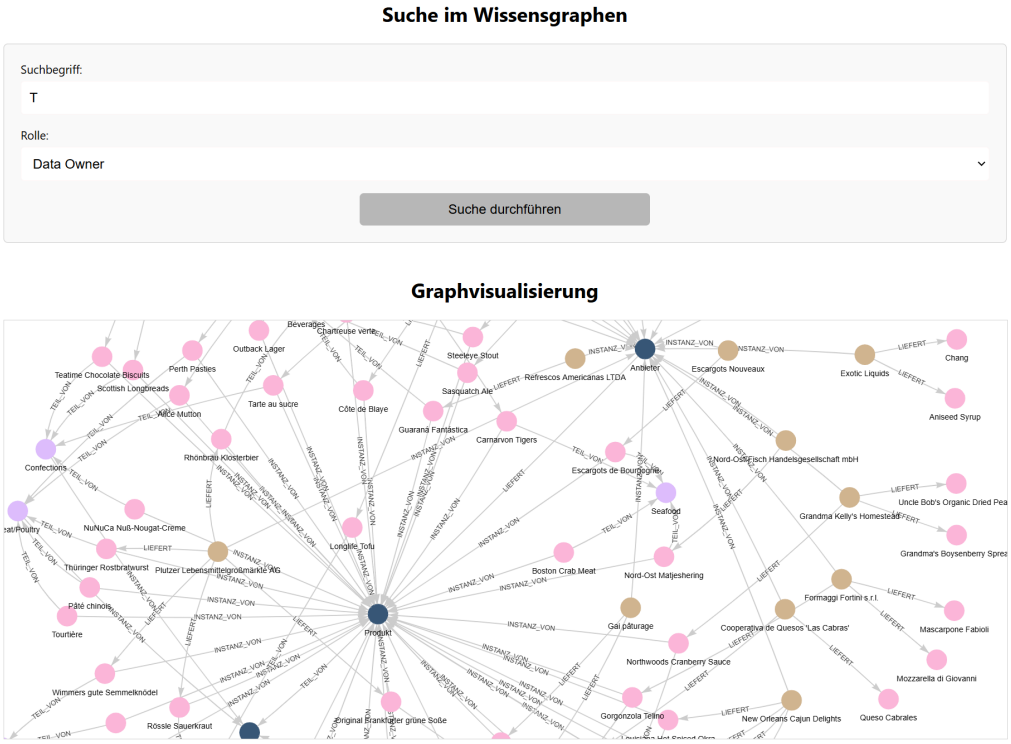


Abbildung 6.7: Möglichkeit zur Durchsuchung des Wissensgraph.

Der Mehrwert dieser Arbeit liegt in der Erstellung und Nutzung von Wissensgraphen zur Vermittlung von Daten. Da bereits einige Arbeiten, wie im Abschnitt 3 erörtert, die Nutzung von Wissensgraphen zur Verbesserung der Suche behandelt haben, wurde die Suchfunktion innerhalb des Graphen in dieser Arbeit nur kurz angeschnitten. Im Allgemeinen kann für die Durchführung der Suche jeder beliebige Algorithmus verwendet werden.

6.3.5 Einfluss der Benutzerrollen

Wie im vorherigem Unterkapitel beschrieben, muss neben dem Suchbegriff auch noch die Rolle des Benutzers eingegeben werden. Wie auch bei der Benutzereingabe ist dieser Schritt bei normalen Anwendungen nicht nötig, da durch die Anmeldung die Rolle des Benutzers gegeben ist. Die Rolle ist von Relevanz, um den Zugriff auf die verschiedenen Ebenen abzubilden, welche in Kapitel 5.1 beschrieben wurden. Nicht alle Rollen dürfen auf alle Ebenen zugreifen, weshalb der Graph in der Suche aus Kapitel 5.2.2 je nach Rolle nur eingeschränkt Ergebnisse zeigen darf.

Um den Zugriff auf die verschiedenen Ebenen zu modellieren, wurden die Labels eingeführt. Jeder implementierte Datensatz, also jeder Knoten, hat ein Label. Diese Labels können anhand der verschiedenen Ebenen unterteilt werden:

- **Domänenebene**: *Oberkategorie*, *Produkt*, *Bedarf*, *Mitarbeiter*, *Anbieter*, *Kategorie* und *GeschäftlicheMetadaten*
- **Datenproduktebene**: *Datenprodukt*, *Bedarf*, *GeschäftlicheMetadaten*
- **Datensatzreferenzebene**: *Datensatzreferenz*, *OperativeMetadaten* und *TechnischeMetadaten*

Wie bereits in Kapitel 5.1.1 erläutert, können Metadaten auf allen Ebenen des Wissensgraphen vorkommen. Allerdings treten verschiedene Arten von Metadaten vermehrt in bestimmten Ebenen auf. So finden sich geschäftliche Metadaten hauptsächlich in der Domänenebene und der Datenproduktebene, da diese Ebenen für die Kontextualisierung und Bedeutung der Daten zuständig sind. Technische Metadaten hingegen kommen vor allem in der Datensatzreferenzebene vor, da sie Informationen über das Format, die Struktur und die Zugriffsrechte der Daten liefern. Operative Metadaten sind hauptsächlich in der Datensatzreferenzebene angesiedelt, da sie detaillierte Informationen zur Data Lineage enthalten.

```
const isRoleRelevant = (label) => {
       if (role === 'Data Scientist') {
           return ['Produkt', 'Bedarf', 'Mitarbeiter', 'Bestellung', 'Oberkategorie',
           'Anbieter', Kategorie','GeschäftlicheMetadaten','OperativeMetadaten']
           .includes(label);
       }
       return true;
   };
```

Listing 6.17: Überprüfung der Rolle und Filterung der Knoten.

In dem erstellten Skript werden zunächst alle Knoten angezeigt die den Suchbegriff enthalten. Im Anschluss werden basierend auf den Rollen die Beziehungen und die verbundenen Knoten gefiltert. Dafür wurde die im Listing 6.17 erstellte Bedingung eingefügt. Diese überprüft den eingegebenen Wert für die Rolle, und lässt die Knoten anzeigen, die die korrekten Label enthalten. In dem Beispiel darf der Data Scientist nur auf die oberen beiden Ebenen zugreifen. Data Stewards und Data Owner dürfen auf alle Daten zugreifen, weshalb diese keine Einschränkung haben.

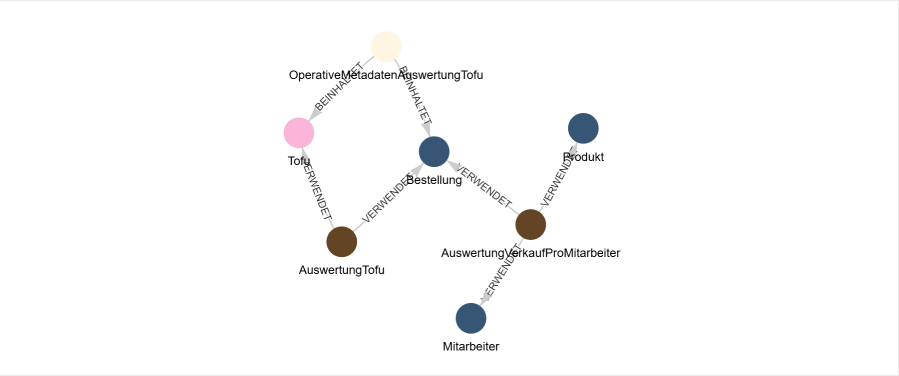


Abbildung 6.8: Sicht auf Graphen aus der Perspektive eines Data Scientists.

Abbildung 6.8 zeigt das Ergebnis einer Suche nach dem Begriff *Auswertung*. Es werden die in Kapitel 6.2.3 erstellten Datenprodukte angezeigt, zusammen mit den erstellten Beziehungen auf die Domänenebene. Wird die Rolle auf Data Owner geändert, erscheint der in Abbildung 6.9 dargestellte Graph. Hier sind ebenfalls die Auswertungsknoten, sowie die Beziehungen zu der Domänenebene zu sehen. Zusätzlich kann jedoch auch die Verbindung zur Datensatzreferenzebene eingesehen werden. Diese Beziehungen und der zusätzliche Knoten werden angezeigt, da der Data Owner Zugriff auf alle Ebenen hat und daher alle Knoten und Verbindungen abfragen kann.

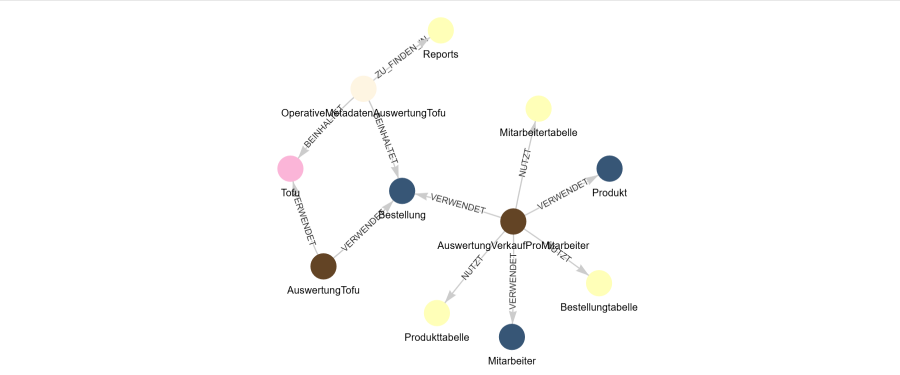
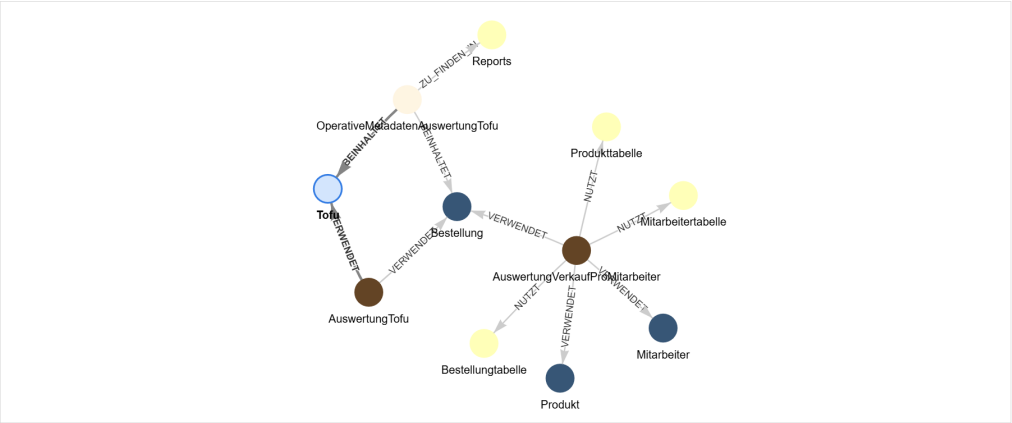


Abbildung 6.9: Sicht auf Graphen aus der Perspektive eines Data Owner.

6.3.6 Attribut von Knoten

Wie in Kapitel 5.2.2 beschrieben, besteht ein Vorteil bei der Nutzung eines Wissengraphen darin, dem Benutzer Beziehungen zwischen verschiedenen Knoten im Wissensgraphen aufzuzeigen, um Datenkonsumenten gezielt auf zusammengehörige Entitäten hinzuweisen. Somit haben die Nutzer das Bedürfnis, die Inhalte eines Knotens einsehen zu können, um festzustellen, ob dieser für ihre Zwecke relevant ist. Hierfür ist eine umfassendere Beschreibung der Knoten erforderlich. Die anzuzeigenden Attribute können individuell festgelegt werden, um sicherzustellen, dass die Darstellung sowohl den datenschutzrechtlichen Anforderungen entspricht als auch den Nutzern die Möglichkeit bietet, relevante neue Knoten zu entdecken.



Attribute des ausgewählten Knotens:

supplierID:	6
productID:	14
productName:	Tofu
categoryID:	7

Abbildung 6.10: Attribute des ausgewählten Knotens *Tofu*.

Wie in Abbildung 5.3.1 dargestellt, werden die Attribute eines Knotens angezeigt, sobald dieser in der Visualisierung vom Benutzer ausgewählt wird. Im gezeigten Beispiel handelt es sich um den Knoten mit dem Namen *Tofu*, welcher nach der Auswahl visuell durch eine blaue Hinterlegung hervorgehoben wird. Unterhalb der Visualisierung erscheint eine Tabelle, die die dazugehörigen Attribute des ausgewählten Knotens übersichtlich darstellt. Damit die Knoten ausgewählt werden können wurde in dem oben gezeigten Code ein Event-Listener eingebaut, welcher bei einem Klick auf einen Knoten ausgelöst wird und die Attribute des Knotens anzeigt.

7 Diskussion

In diesem Kapitel werden das entwickelte Konzept, sowie die durchgeführte Implementierung kritisch beleuchtet. Zunächst erfolgt eine Evaluation des Konzepts und des Prototyps. Anschließend werden Limitierungen der Arbeit reflektiert.

7.1 Evaluation

Wie zu Beginn von Kapitel 5 beschrieben, konnten nicht alle Anforderungen bei der Erstellung des Konzepts berücksichtigt wird. Aus diesem Grund konzentriert sich diese Arbeit auf diejenigen Anforderungen, die den Fokus auf die Vermittlung von Daten zwischen Datenproduzenten und Datenkonsumenten legt. In Tabelle 7.1 sind diese Anforderungen aufgelistet, gemeinsam mit dem Grad der Erfüllung im Konzept und der Umsetzung im Prototypen. Der Grad der Erfüllung ist in fünf Abstufungen unterteilt. An erster Stelle steht *vollständig erfüllt*, was bedeutet, dass die Anforderung vollständig erfüllt werden konnte. Darauf folgt *weitestgehend erfüllt*, wobei der Großteil der Anforderung erfolgreich abgeschlossen wurde, jedoch kleinere, nicht-kritische Aspekte noch fehlen. Die Stufe *teilweise erfüllt* beschreibt eine Umsetzung, bei der wesentliche Teile zwar realisiert wurden, jedoch wichtige Elemente ausstehen oder deutliche Abweichungen erkennbar sind. Wird eine Anforderung *rudimentär erfüllt*, so handelt es sich um erste Schritte, die jedoch noch in einem frühen, wenig ausgearbeiteten oder unzureichenden Stadium verbleiben. Schließlich steht *nicht erfüllt* für den Fall, dass entweder keine Anforderung eingeleitet wurden oder die Umsetzung vorzeitig beendet wurde. Im Folgenden werden diese Anforderungen einzeln reflektiert und eingestuft.

	Anforderung	**Umsetzung in Konzept**	**Umsetzung in Prototyp**
A1	Datenzugriffsverwaltung	Vollständig erfüllt	Weitestgehend erfüllt
A4	Metadatenmanagement	Vollständig erfüllt	Vollständig erfüllt
A6	Datenbenachrichtigung	Vollständig erfüllt	Rudimentär erfüllt
A10	Kommunikation	Vollständig erfüllt	Weitestgehend erfüllt
A12	Datenzugriffsanfragen	Vollständig erfüllt	Weitestgehend erfüllt
A13	Intelligente Suche	Teilweise erfüllt	Rudimentär erfüllt
A15	Rollenverteilung	Vollständig erfüllt	Teilweise erfüllt
A16	Datenprodukte	Teilweise erfüllt	Teilweise erfüllt

Tabelle 7.1: Übersicht über die Umsetzung der Anforderungen im Konzept und Prototyp.

Im Kontext der Datenvermittlung sind sowohl eine effiziente Datenzugriffsverwaltung (*A1*) als auch eine transparente Regelung für Datenzugriffsanfragen (*A12*) von Bedeutung, um den Bedarf an neuen Daten nachvollziehbar zu erfassen. Die Realisierung dieser Anforderungen erfolgt über

Relationen, die im Bedarfsknoten definiert werden. Darüber initiiert ein Datenkonsument eine Datenzugriffsanfrage an den entsprechenden Datenproduzenten. Der Datenproduzent wird durch eine Benachrichtigung über die Anfrage informiert und kann anschließend den Datenzugriff steuern. Dies umfasst zwei mögliche Optionen: Zum einen die Bereitstellung der Daten, wenn alle erforderlichen Bedingungen erfüllt sind oder die Klärung mit dem Datenkonsumenten, um beispielsweise sicherzustellen, dass die Datenbereitstellung keine Datenschutzverletzungen verursacht. Zum Anderen die Ablehnung der Anfrage, indem auf eine Datenbereitstellung verzichtet wird.
Im Prototypen konnte diese Funktionalität weitestgehend erfüllt werden. Die Erstellung eines Bedarfsknotens generiert eine Datenzugriffsanfrage, die den Datenproduzenten informiert, dass eine Bereitstellung erforderlich ist. Der Entscheidungsprozess zur Handhabung der Anfrage, einschließlich der Bereitstellung oder Ablehnung der Daten, ist jedoch im Prototypen nicht vollständig umgesetzt. Zwar kann der Datenproduzent den Status des Bedarfsknotens aktualisieren, beispielsweise um anzugeben, dass die Bereitstellung in Bearbeitung ist oder abgeschlossen wurde; Allerdings wird die Möglichkeit, eine Anfrage abzulehnen, nicht abgebildet. Diese Entscheidung und die damit verbundene Kommunikation finden hier außerhalb des Bedarfsknotens statt und sind somit nicht in den Prototypen integriert.

Als nächste Anforderung wurde das Metadatenmanagement definiert. In Kapitel 5.1.1 wird eine Metadatenebene eingeführt, die eine zentrale Rolle beim Aufbau des Wissensgraphen spielt. Diese Ebene dient sowohl Datenproduzenten als auch Datenkonsumenten dazu, ein besseres Verständnis der Daten zu erlangen. Sie stellt geschäftliche Metadaten bereit, die die Bedeutung und den Kontext der Daten erläutern, sowie technische Metadaten, die bei der Bereitstellung der Daten berücksichtigt werden müssen. Darüber hinaus unterstützt sie die Datenvermittlung durch die Dokumentation der Data Lineage über operative Metadaten.
Im Prototypen wurde diese Funktionalität durch die Einführung spezifischer Labels für jede Art von Metadaten umgesetzt. Diese Labels ermöglichen es, die Metadaten den entsprechenden Ebenen im Wissensgraphen zuzuordnen und somit die Struktur und Übersichtlichkeit des Graphen zu verbessern. Jedoch mussten die Metadaten manuell angelegt und entworfen werden, da der verwendete Northwind-Datensatz keine Metadaten zur Verfügung stellt.

Datenproduzenten sollen eine Benachrichtigung erhalten, sobald sie aufgefordert werden, Datensätze bereitzustellen. Dieser Aspekt wird durch *A6*, die Datenbenachrichtigung, adressiert. Im Konzept wird hierfür ein Benachrichtigungssystem vorgeschlagen, das über Trigger funktioniert. Dieses System informiert Benutzer, sobald sie durch eine Relation mit einem Bedarfsknoten verbunden werden.
Im Prototypen konnte diese Funktionalität jedoch nur eingeschränkt umgesetzt werden. Da der Prototyp aus Gründen des hohen Aufwands kein Backend enthält, das eine Authentifizierung und Verwaltung verschiedener Benutzer ermöglicht, wurde als Alternative die Funktion *Benutzereingabe* implementiert. Dabei gibt ein Benutzer seine ID an und erhält daraufhin eine Übersicht der Bedarfsknoten, die ihm zugewiesen wurden oder die er selbst erstellt hat. Allerdings ersetzt diese Lösung keine automatische Benachrichtigung. In einem realen Unternehmenskontext wird der Wissensgraph über eine Anwendung aufgerufen, in der sich die Akteure anmelden können. Dadurch kann die manuelle Eingabe einer Benutzer-ID entfallen. Zudem ist es möglich, im Backend automatisierte Benachrichtigungen, beispielsweise per E-Mail, zu versenden, sobald eine Zuweisung zu einem Bedarfsknoten erfolgt. Der volle Umfang dieser Funktionalität, insbesondere die Integration eines Backends und eines Benachrichtigungssystems, wurde im Prototyp nicht umgesetzt. Dennoch konnte durch die implementierte Benutzereingabe die Kernidee des Konzepts demonstriert werden.

Als nächstes folgt Anforderung *A10*, die „Kommunikation“. Ein effektiver Austausch zwischen Datenproduzenten und Datenkonsumenten ist unerlässlich, um eine verbesserte Vermittlung von Daten zu ermöglichen. Dafür wird im Konzept die Verwendung eines Wissensgraphen und die Einführung eines sogenannten Bedarfsknotens in diesem vorgeschlagen. Dieser Knoten ermöglicht es, zentrale Attribute wie Datenqualität, zeitliche Aspekte sowie allgemeine Beschreibungen des Datenbedarfs zu dokumentieren. Durch die zentrale Verwaltung dieser Informationen wird die Kommunikation zwischen den beteiligten Akteuren erleichtert. Ein weiterer Vorteil des Bedarfsknotens besteht in der Möglichkeit, Relationen zu Datenproduzenten direkt im System zu hinterlegen. Dadurch entfällt die Notwendigkeit, jeden Datenproduzenten individuell zu kontaktieren. Stattdessen genügt es, den Bedarfsknoten einmalig zu erstellen. Über diesen Knoten können die Datenproduzenten nachvollziehen, wem sie die Daten bereitstellen sollen und welche weiteren Produzenten beteiligt sind. Dies erleichtert die Koordination und ermöglicht eine direkte Abstimmung untereinander. Gleichzeitig können Datenkonsumenten den Status der Datenbereitstellung einsehen, ohne individuell bei den Data Ownern nachfragen zu müssen. Ein Verbesserungsvorschlag betrifft die Flexibilität bei der Anlage von Bedarfsknoten, wie sie unter *Einschränkungen* in Kapitel 5.3.1 beschrieben wird. Wenn der Ersteller eines Bedarfsknotens nicht weiß, welche Datenproduzenten den benötigten Datensatz bereitstellen, soll der Bedarfsknoten dennoch erstellt werden können. Das System kann dann mithilfe bestehender Relationen – wie in Abbildung 5.6 illustriert – automatisch die relevanten Datenproduzenten identifizieren. Derzeit müssen in der Prototyp-Umsetzung die Datenproduzenten allerdings manuell hinzugefügt werden. Der verwendete Northwind-Datensatz enthält keine Verbindungen zwischen Akteuren und den einzelnen Produkten, welche nötig sind, um die Relationen automatisch hinterlegen zu können, weshalb dies im Prototypen nicht umgesetzt wurde.

Ein weiterer zentraler Aspekt ist die „intelligente Suche“, wie sie in Anforderung *A13* beschrieben wird. In Kapitel 5.2.2 wurde das Konzept der Suche dargestellt. Sie spielt eine wesentliche Rolle bei der Vermittlung von Daten zwischen Datenproduzenten und Datenkonsumenten, indem sie die Suche und den Zugriff auf relevante Informationen optimiert.
Ein Wissensgraph aggregiert Daten aus verschiedenen Quellen und verknüpft sie basierend auf gemeinsamen Entitäten oder Attributen. Dies ermöglicht es, Daten aus unterschiedlichen Kontexten semantisch miteinander zu verknüpfen. Datenproduzenten profitieren davon, dass ihre Daten nicht isoliert betrachtet werden, sondern nahtlos mit anderen relevanten Informationen kombiniert werden können. Dies verbessert die Auffindbarkeit der Daten für potenzielle Konsumenten. Gleichzeitig erhalten Datenkonsumenten einen vollständigen Überblick über verteilte Datenquellen, ohne manuell mehrere Systeme durchsuchen oder verschiedene Datenstrukturen verstehen zu müssen. Die intelligente Suche im Wissensgraphen fungiert dabei als zentrale Vermittlungsinstanz, die Konsumenten nicht nur auf die Daten verweist, sondern auch kontextuelle Informationen und Beziehungen liefert. Dadurch wird die Datenvermittlung sowohl effizienter als auch präziser gestaltet, was die Interaktion zwischen Produzenten und Konsumenten vereinfacht.
Wie in Kapitel 6.3.4 beschrieben, wurde die Suche im Rahmen der Implementierung lediglich in eingeschränkter Form realisiert. Da bereits zahlreiche wissenschaftliche Arbeiten den Mehrwert von Wissensgraphen für die Verbesserung von Suchfunktionen untersucht haben, liegt dieser Aspekt nicht im Fokus dieser Arbeit. Aus diesem Grund wurde die Suche rudimentär erfüllt, ohne die Anwendung komplexer Algorithmen. Stattdessen wurde ein einfacher Wortabgleich implementiert, der die grundlegende Idee einer Suchfunktion veranschaulicht.

Ebenfalls wichtig ist die Rollenverteilung (*A15*). Die Zuweisung spezifischer Rollen an die beteiligten Akteure trägt zur Vereinfachung und Strukturierung der Datenzugriffsverwaltung bei. Im Konzept wurde die Rollenverteilung, wie in Kapitel 5.1.4 beschrieben, umgesetzt.
In der Implementierung wurden die Rollen auf vier zentrale Akteursgruppen beschränkt: Data Scientist, Data Engineer, Data Owner und Data Steward. Die Rollenverteilung diente in erster Linie dazu, den Zugriff auf die verschiedenen Ebenen des Wissensgraphen zu beschränken. Das trägt zur besseren Übersichtlichkeit bei und erleichtert die Suche innerhalb des Wissensgraphen. Innerhalb eines Unternehmens existieren jedoch häufig interne Berechtigungsgruppen und -rollen, die den Datenzugriff regeln. Diese wurden im Rahmen des Prototyps nicht modelliert, da keine entsprechenden Daten vorliegen. In der Theorie kann das durch die Einführung eines zusätzlichen Attributs für jeden Akteur umgesetzt werden, welches die entsprechenden Berechtigungsrollen beschreibt. Eine solche Erweiterung unterstützt die Datenproduzenten bei der Entscheidung, ob sie Daten bereitstellen können oder nicht. Das kann insbesondere dann hilfreich sein, wenn der Zugang zu sensiblen oder geschützten Daten geregelt werden muss.

A16 (Datenprodukte) wurde mit der Erstellung eines Bedarfsknotens teilweise erfüllt. Das Endergebnis eines Bedarfsknotens auf der Datenproduktebene ist ein fertiges Datenprodukt. Durch die Generierung eines Bedarfsknotens wird die Definition und Erstellung dieses Datenprodukts vorgenommen. Allerdings wurde die eigentliche Bereitstellung dieser Datenprodukte im Rahmen der Arbeit nicht behandelt, weshalb diese Anforderung nur teilweise erfüllt ist.
Das Selbe gilt für die Implementierung. Durch die Möglichkeit der Erstellung eines Bedarfsknotens wurde die Anforderung ebenfalls teilweise erfüllt. Zwar wurde die Erstellung des Bedarfsknotens erfolgreich realisiert, jedoch ist die konkrete Vermittlung der daraus resultierenden Datenprodukte unberücksichtigt geblieben.

7.2 Limitation

Nachdem alle Anforderungen evaluiert wurden, werden nun die wissenschaftlichen Limitationen der Arbeit betrachtet. Dabei wird untersucht, inwieweit die entwickelten Konzepte universell anwendbar sind und welche Einschränkungen bestehen. Zudem wird erläutert, inwieweit die Evaluation aussagekräftig ist und welche Faktoren ihre Aussagekraft möglicherweise begrenzen.

In einem Data Lake werden, wie in Kapitel 2.1.1 beschrieben, strukturierte, semi-strukturierte und unstrukturierte Daten gespeichert und verarbeitet. Der im Prototyp verwendete Datensatz ist jedoch strukturiert, sodass in dieser Arbeit ausschließlich strukturierte Daten genutzt wurden. Es existieren bereits Ansätze, wie unstrukturierte Daten in Wissensgraphen integriert werden können. So beschreibt beispielsweise der Artikel [May24], wie ein Wissensgraph aus unstrukturiertem Text mithilfe von Large Language Models (LLMs) erstellt werden kann. Obwohl solche Verfahren vielversprechend sind, lag der Schwerpunkt dieser Arbeit nicht auf der Erstellung des Wissensgraphen, sondern auf der Modellierung und Nutzung von Bedarfsknoten. Die Einbindung unstrukturierter Daten wurde daher bewusst nicht berücksichtigt.

Des Weiteren wurde die Skalierbarkeit in der vorliegenden Arbeit nicht berücksichtigt. Im Prototyp wurde mit einer begrenzten Datenmenge gearbeitet, wohingegen Unternehmen in der Regel wesentlich umfangreichere Datensätze verwalten. Dies impliziert, dass die Ergebnisse hinsichtlich der Skalierbarkeit nur eingeschränkt auf reale Anwendungsszenarien übertragbar sind.

Darüber hinaus wurden Aspekte wie Performance unter realen Bedingungen sowie Datensicherheit und Datenschutz nicht mit einbezogen. Auch die Integration in bestehende Systeme wurde nicht berücksichtigt. Diese Einschränkungen machen eine direkte Übertragung der Ergebnisse in die Praxis schwierig und unterstreichen den Bedarf an weiterführender Forschung.

8 Fazit und Ausblick

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst. Zunächst wird ein Überblick über die zentralen Inhalte und Erkenntnisse gegeben. Es werden die Inhalte der Arbeit zusammengefasst, es wird ein Fazit gezogen und es wird ein Ausblick auf mögliche zukünftige Arbeiten gegeben.

8.1 Zusammenfassung

Die stetig wachsende Menge verfügbarer Daten erfordert effiziente Methoden zur Datenerhebung, -analyse und -verwaltung. Insbesondere umfangreiche Datenmengen müssen erfasst, verarbeitet und strukturiert organisiert werden, wofür sich Datenplattformen als zentrale Lösung etabliert haben. Eine der größten Herausforderungen solcher Plattformen liegt in der effektiven Vermittlung von Daten zwischen Datenproduzenten und -konsumenten. In dieser Arbeit wurde die Nutzung von Wissensgraphen zur Bewältigung dieser Aufgabe untersucht.

Zunächst wurde basierend auf den Aufgaben der Datenproduzenten und -konsumenten Anforderungen von Datenkonsumenten und -produzenten abgeleitet. Basierend auf der Zielsetzung der Arbeit wurden anschließend die passendsten Anforderungen ausgewählt. Auf dieser Grundlage wurde ein Konzept erstellt, das die Nutzung eines Wissensgraphen zur Vermittlung von Daten beinhaltet. Im Zentrum dieses Konzepts steht der Bedarfsknoten. Dieser ermöglicht es den Datenkonsumenten, ihren Bedarf an Daten direkt im Wissensgraphen zu visualisieren. Gleichzeitig bietet er den Datenproduzenten eine zentrale Anlaufstelle, um alle relevanten Informationen zur Bereitstellung der angeforderten Daten zu erhalten.

Zur Evaluation wurde eine prototypische Anwendung entwickelt, die einen Wissensgraphen enthält. Diese erlaubt es, Bedarfsknoten zu erstellen und datenbezogene Anforderungen zu formulieren. Gleichzeitig bietet sie Funktionen, um auf diese Bedarfsknoten zu reagieren und so die Datenvermittlung zwischen Konsumenten und Produzenten zu unterstützen.

Der Prototyp hat gezeigt, dass das entwickelte Konzept technisch umsetzbar ist und die Einführung von Bedarfsknoten in einem Wissensgraphen prinzipiell funktioniert.

8.2 Fazit

Das Ziel dieser Arbeit wurde vollständig erreicht. Es wurde ein Konzept entwickelt, welches die Nutzung eines Wissensgraphen zur Vermittlung von Daten zwischen Datenproduzenten und -konsumenten beschreibt. Dabei wurden die zuvor ermittelten Anforderungen systematisch integriert und thematisch aufbereitet. Wie in Kapitel 7 dargelegt, konnte der Großteil dieser Anforderungen

im Konzept erfolgreich berücksichtigt werden. Die prototypische Implementierung hat zudem verdeutlicht, dass die grundlegende Idee des Konzepts umsetzbar ist, auch wenn nicht alle Aspekte im Detail realisiert werden konnte. Insgesamt hat sich das Konzept als praktikabel und zielführend erwiesen.

Trotzdem zeigte die Implementierung auch Limitationen auf, insbesondere bezüglich der Skalierbarkeit sowie hinsichtlich der Performance und der Integration in bestehende Systeme. Diese Einschränkungen identifizieren potenzielle Verbesserungsbereiche und dienen als Ausgangspunkt für zukünftige Weiterentwicklungen.

Die Ergebnisse dieser Arbeit bieten einen neuen Ansatz, um die Vermittlung von Daten zwischen Konsumenten und Produzenten innerhalb eines Unternehmens zu verbessern. Durch die Verwendung von Wissensgraphen wird nicht nur die Vermittlung zwischen Datenproduzenten und -konsumenten vereinfacht, sondern auch eine tiefere Vernetzung und Kontextualisierung der Daten ermöglicht. Dies fördert die Wiederverwendbarkeit bestehender Daten und reduziert Redundanzen bei der Datenerhebung oder -speicherung. Dieses Konzept stellt somit einen vielversprechenden Beitrag zur Optimierung datengetriebener Prozesse in Unternehmen dar. Langfristig kann die Implementierung eines solchen Ansatzes die Effizienz und Innovationskraft von Unternehmen stärken, indem sie datengetriebene Projekte vereinfacht.

8.3 Ausblick

Das entwickelte Konzept bieten eine vielversprechende Grundlage für die Vermittlung von Daten zwischen Produzenten und Konsumenten mithilfe eines Wissensgraphen. Dennoch konnten im Rahmen dieser Arbeit nicht alle erstellten Anforderungen vollständig berücksichtigt werden. Eine mögliche Weiterentwicklung ist, in zukünftigen Arbeiten die verbleibenden Anforderungen systematisch in das Konzept zu integrieren. Besonders wichtig ist dabei unter Anderem *A11 (Verknüpfung von Datensätzen)*. Sobald der Bedarfsknoten erstellt ist, müssen die Daten für die Datenkonsumenten zugänglich gemacht werden. Dafür ist es notwendig, Daten aus verschiedenen Quellen und von unterschiedlichen Data Ownern zu kombinieren. Hierbei muss das System unterstützen. Auf diese Weise können sowohl die Bedürfnisse der Datenproduzenten als auch der Datenkonsumenten umfassender abgedeckt werden.

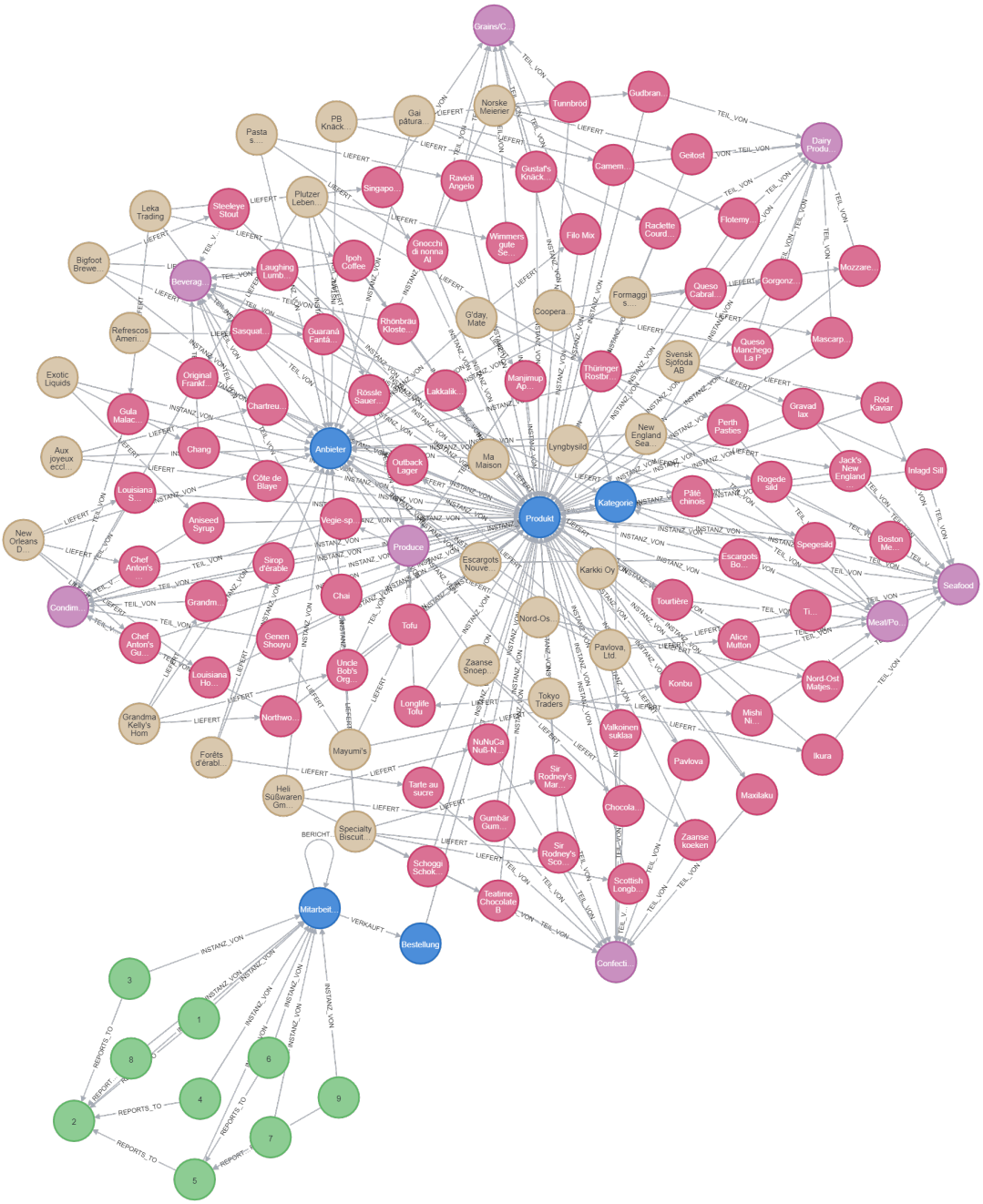
Darüber hinaus birgt die Implementierung Optimierungspotenzial. Die prototypische Umsetzung hat gezeigt, dass das Konzept grundsätzlich funktionsfähig sind. Jedoch können künftige Arbeiten die Implementierung erweitern, um alle Anforderungen vollständig zu realisieren und die Anwendung zu verfeinern. Insbesondere die Integration fortgeschrittener Funktionen wie der automatisierten Verknüpfung von Bedarfsknoten oder die Optimierung der Benutzerfreundlichkeit können das Konzept weiter verbessern.

Ein weiterer Ansatz für zukünftige Arbeiten besteht in der Erprobung des Konzepts im realen Unternehmenskontext. Hierbei kann untersucht werden, wie sich die Nutzung eines Wissensgraphen in einem echten DLH mit tatsächlichen Unternehmensdaten bewährt. Dies liefert nicht nur praxisnahe Einblicke in die Funktionalität des Konzepts, sondern auch dazu beitragen, mögliche Herausforderungen, beispielsweise in Hinblick auf die Integration in bestehende IT-Systeme oder die Handhabung sensibler Daten, zu identifizieren und zu lösen.

einer Nutzerstudie, um zu evaluieren, inwiefern Benutzer mit unterschiedlichen Rollen von dem entwickelten Konzept profitieren. Eine solche Untersuchung kann wertvolle Erkenntnisse darüber liefern, ob und in welchem Maße das Konzept die verschiedenen Nutzerbedürfnisse adressiert und welche Anpassungen notwendig sind, um eine optimale Unterstützung für alle Beteiligten zu gewährleisten.

Zudem bietet es sich an, die Skalierbarkeit des Wissensgraphen zu analysieren. Während im Prototyp mit einer begrenzten Datenmenge gearbeitet wurde, speichern Unternehmen in der Regel mehr Daten. Eine Untersuchung, wie sich der Wissensgraph mit einer großen Datenmenge verhält, kann wichtige Erkenntnisse zur Performance und Effizienz liefern.

9 Anhang



Anhang A1: Domänenebene in Neo4j-Browser

Literaturverzeichnis

[AGXZ21] M. Armbrust, A. Ghodsi, R. Xin, M. Zaharia. „Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics“. In: *Proceedings of CIDR*. Bd. 8. 2021, S. 28 (zitiert auf S. 3–6).

[Apa19] ApacheAtlas. *Atlas-Autorisierungsmodell*. 2019. URL: https://atlas.apache.org/2.0.0/Atlas-Authorization-Model.html (zitiert auf S. 14).

[Atl24] Atlan. *Apache Atlas: Origins, Architecture, Capabilities, Installation, Alternatives Comparison*. 2024. URL: https://atlan.com/what-is-apache-atlas/ (zitiert auf S. 13).

[Bai24] J. Baig. *Was ist ein Datenkatalog? Funktionen, Best Practices und Vorteile*. 2024. URL: https://www.astera.com/de/type/blog/data-catalog/ (zitiert auf S. 13).

[BGK21] E. Begoli, I. Goethert, K. Knight. „A lakehouse architecture for the management and analysis of heterogeneous data for biomedical research and mega-biobanks“. In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE. 2021, S. 4643–4651 (zitiert auf S. 6).

[Bis25] Bissantz. *Daten in Echtzeit bereitstellen*. 2025. URL: https://www.bissantz.de/know-how/data-warehousing/daten-in-echtzeit-bereitstellen/ (zitiert auf S. 32).

[BSN+09] J. v. Brocke, A. Simons, B. Niehaves, B. Niehaves, K. Reimer, R. Plattfaut, A. Cleven. „Reconstructing the giant: On the importance of rigour in documenting the literature search process“. In: (2009) (zitiert auf S. 14).

[Cao17] L. Cao. „Data science: challenges and directions“. In: *Communications of the ACM* 60.8 (2017), S. 59–68 (zitiert auf S. 20, 21).

[Clo24] G. Cloud. *Data Catalog overview*. 2024. URL: https://cloud.google.com/data-catalog/docs/concepts/overview (zitiert auf S. 6).

[Cod24] V. S. Code. *Code faster with AI*. 2024. URL: https://code.visualstudio.com/ (zitiert auf S. 43).

[DAM17] I. DAMA. *DAMA-DMBOK2 Data Management Body of Knowledge, 2nd Editon*. 2017 (zitiert auf S. 6).

[Dat24a] Databricks. *What is a Data Lakehouse?* 2024. URL: https://www.databricks.com/glossary/data-lakehouse (zitiert auf S. 4–6).

[Dat24b] DataGalaxy. *Demystifying data catalogs: A comprehensive guide to organizing your data*. 2024. URL: https://www.datagalaxy.com/en/blog/demystifying-data-catalogs-a-comprehensive-guide-to-organizing-your-data/ (zitiert auf S. 7).

[Dat24c] DataInstitute. *Data Steward*. 2024. URL: https://www.datainstitute.io/glossar/data-steward (zitiert auf S. 19).

[dat25] databricks. *A.C.I.D. properties: Atomicity, Consistency, Isolation, and Durability*. 2025. URL: https://www.databricks.com/glossary/acid-transactions#:~:text=ACID%20is%20an%20acronym%20that,operations%20are%20called%20transactional%20systems. (zitiert auf S. 6).

[DGM+18] C. Diamantini, P. L. Giudice, L. Musarella, D. Potena, E. Storti, D. Ursino. „A new metadata model to uniformly handle heterogeneous data lake sources“. In: *New Trends in Databases and Information Systems: ADBIS 2018 Short Papers and Workshops, AI* QA, BIGPMED, CSACDB, M2U, BigDataMAPS, ISTREND, DC, Budapest, Hungary, September, 2-5, 2018, Proceedings 22*. Springer. 2018, S. 165–177 (zitiert auf S. 6).

[Dha13] V. Dhar. „Data science and prediction“. In: *Communications of the ACM* 56.12 (2013), S. 64–73 (zitiert auf S. 20, 21).

[DS21] H. Dibowski, S. Schmid. „Using knowledge graphs to manage a data lake“. In: *INFORMATIK 2020*. Gesellschaft für Informatik, Bonn. 2021, S. 41–50 (zitiert auf S. 15).

[DSS+20] H. Dibowski, S. Schmid, Y. Svetashova, C. Henson, T. Tran. „Using Semantic Technologies to Manage a Data Lake: Data Catalog, Provenance and Access Control.“ In: *SSWS@ ISWC*. Athen. 2020, S. 65–80 (zitiert auf S. 2, 14).

[esp25] esphere. *Trigger*. 2025. URL: https://e-sphere.ch/de/glossar/trigger (zitiert auf S. 32).

[EW16] L. Ehrlinger, W. Wöß. „Towards a definition of knowledge graphs.“ In: *SEMANTiCS (Posters, Demos, SuCCESS)* 48.1-4 (2016), S. 2 (zitiert auf S. 10).

[FAK+18] R. C. Fernandez, Z. Abedjan, F. Koko, G. Yuan, S. Madden, M. Stonebraker. „Aurum: A data discovery system“. In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE. 2018, S. 1001–1012 (zitiert auf S. 15).

[Fiv23] FiveTeams. *Der Beruf des Data Scientists: Aufgaben und Voraussetzungen*. 2023. URL: https://www.fiveteams.com/ratgeber/data-scientist-aufgaben (zitiert auf S. 21).

[GGH+19] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang. „Leveraging the data lake: current state and challenges“. In: *Big Data Analytics and Knowledge Discovery: 21st International Conference, DaWaK 2019, Linz, Austria, August 26–29, 2019, Proceedings 21*. Springer. 2019, S. 179–188 (zitiert auf S. 6, 8).

[GGH+20] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang. „A zone reference model for enterprise-grade data lake management“. In: *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE. 2020, S. 57–66 (zitiert auf S. 8, 9, 28).

[Göl24] S. Göllner. *Datenanalyse – Wie du Daten in wertvolle Informationen verwandelst*. 2024. URL: https://bimanu.de/blog/datenanalyse/ (zitiert auf S. 1).

[GP21] G. Guizzardi, H. A. Proper. „On understanding the value of domain modeling“. In: *CEUR workshop proceedings*. Bd. 2835. Rheinisch Westfälische Technische Hochschule. 2021, S. 51–62 (zitiert auf S. 26).

[Grö21] C. Gröger. „There is no AI without data“. In: *Communications of the ACM* 64.11 (2021), S. 98–108 (zitiert auf S. 1, 17–21).

[Grö22] C. Gröger. „Industrial analytics–An overview“. In: *it-Information Technology* 64.1-2 (2022), S. 55–65 (zitiert auf S. 3, 6, 8).

[GSM14] C. Gröger, H. Schwarz, B. Mitschang. „The deep data warehouse: link-based integration and enrichment of warehouse data and unstructured content“. In: *2014 IEEE 18th International Enterprise Distributed Object Computing Conference*. IEEE. 2014, S. 210–217 (zitiert auf S. 1).

[HBC+21] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann. „Knowledge Graphs“. In: *ACM Comput. Surv.* 54.4 (Juli 2021). ISSN: 0360-0300. DOI: 10.1145/3447772. URL: https://doi.org/10.1145/3447772 (zitiert auf S. 2, 10).

[He18] J. C. He. *Apache Atlas and JanusGraph – graph-based meta data management*. 2018. URL: https://developer.ibm.com/articles/apache-atlas-and-janusgraph-graph-based-meta-data-management/ (zitiert auf S. 13).

[Hel21] A. Helal. „Data lakes empowered by knowledge graph technologies“. In: *Proceedings of the 2021 International Conference on Management of Data*. 2021, S. 2884–2886 (zitiert auf S. 15).

[HGQ16] R. Hai, S. Geisler, C. Quix. „Constance: An intelligent data lake system“. In: *Proceedings of the 2016 international conference on management of data*. 2016, S. 2097–2100 (zitiert auf S. 16).

[HHAM21] A. Helal, M. Helali, K. Ammar, E. Mansour. „A demonstration of kglac: A data discovery and enrichment platform for data science“. In: *Proceedings of the VLDB Endowment* 14.12 (2021), S. 2675–2678 (zitiert auf S. 15).

[HORB22] T. Hlupić, D. Oreščanin, D. Ružak, M. Baranović. „An overview of current data lake architecture models“. In: *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*. IEEE. 2022, S. 1082–1087 (zitiert auf S. 8, 9).

[HTQ24] S. Hoseini, J. Theissen-Lipp, C. Quix. „A survey on semantic data management as intersection of ontology-based data access, semantic modeling and data lakes“. In: *Journal of Web Semantics* 81 (2024), S. 100819 (zitiert auf S. 14, 28).

[HZ22] A. A. Harby, F. Zulkernine. „From data warehouse to lakehouse: A comparative review“. In: *2022 IEEE International Conference on Big Data (Big Data)*. IEEE. 2022, S. 389–395 (zitiert auf S. 4, 5).

[IBM24a] IBM. *Datenkatalog*. 2024. URL: https://www.ibm.com/de-de/topics/data-catalog (zitiert auf S. 6, 7).

[IBM24b] IBM. *Was ist ein Wissensgraph?* 2024. URL: https://www.ibm.com/de-de/topics/knowledge-graph (zitiert auf S. 7, 10, 29).

[IBM24c] IBM. *Was ist Extrahieren, Laden, Transformieren (ELT)?* 2024. URL: https://www.ibm.com/de-de/topics/elt (zitiert auf S. 5).

[IBM25a] IBM. *Defining user identifiers (UIDs)*. 2025. URL: https://www.ibm.com/docs/en/zos/3.1.0?topic=unix-defining-user-identifiers-uids (zitiert auf S. 34).

[IBM25b] IBM. *What is business intelligence (BI)?* 2025. URL: https://www.ibm.com/think/topics/business-intelligence (zitiert auf S. 4).

[IBM25c] IBM. *What is ELT?* 2025. URL: https://www.ibm.com/think/topics/elt#:~:text=Use%20cases-,What%20is%20ELT%3F,such%20as%20a%20data%20warehouse. (zitiert auf S. 6).

[Ins22] C. Institut. *Job Profile: Data Owner*. 2022. URL: https://insights.controller-institut.at/job-profile-data-owner/#:~:text=Der%20Data%20Owner%20ist%20die,insbesondere%20der%20Finanzdaten%2C%20%C3%BCbernehmen%20kann. (zitiert auf S. 18).

[Ins24] D. Institute. *Data Owner*. 2024. URL: https://www.datainstitute.io/glossar/data-owner#:~:text=Was%20sind%20Aufgaben%20von%20Data,die%20ben%C3%B6tigten%20Daten%20abgerufen%20werden. (zitiert auf S. 18).

[Jan24] JanusGraph. *JanusGraph - Distributed, open source, massively scalable graph database*. 2024. URL: https://janusgraph.org/ (zitiert auf S. 13).

[Kat24] Kato. *Data Catalog Meta data system in Enterprise*. 2024. URL: https://katoengineer.com/posts/data-catalog/ (zitiert auf S. 14).

[KGL+20] E. G. Kalaycı, I. Grangel González, F. Lösch, G. Xiao, A. ul-Mehdi, E. Kharlamov, D. Calvanese. „Semantic integration of Bosch manufacturing data using virtual knowledge graphs“. In: *The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II 19*. Springer. 2020, S. 464–481 (zitiert auf S. 16).

[Kla24] F. Klasen. *Was macht ein Data Scientist?* 2024. URL: https://techminds.de/jobprofile/data-scientist/ (zitiert auf S. 20, 21).

[Kli24] Klipfolio. *What is a Data Catalog?* 2024. URL: https://www.klipfolio.com/resources/data-stack/data-catalog (zitiert auf S. 13).

[Kro22] M. Kromer. *Mapping Data Flows in Azure Data Factory*. Springer, 2022. ISBN: 978-1-4842-8612-8. DOI: https://doi.org/10.1007/978-1-4842-8612-8 (zitiert auf S. 19, 20).

[KZDB16] M. Kim, T. Zimmermann, R. DeLine, A. Begel. „The emerging role of data scientists on software development teams“. In: *Proceedings of the 38th International Conference on Software Engineering*. 2016, S. 96–107 (zitiert auf S. 20).

[LAX+20] B. Lorica, M. Armbrust, R. Xin, M. Zaharia, A. Ghodsi. *What is a Lakehouse?* 2020. URL: https://www.databricks.com/blog/2020/01/30/what-is-a-data-lakehouse.html (zitiert auf S. 6).

[LCL21] P. Luo, J. Chen, J. Li. „Enterprise big data management based on Knowledge Graph“. In: *2021 4th International Conference on Computing and Big Data*. 2021, S. 54–59 (zitiert auf S. 9, 10).

[LL19] S. Luber, N. Litzel. *Was ist ein Dateneigner?* 2019. URL: https://www.bigdata-insider.de/was-ist-ein-dateneigner-a-866465/ (zitiert auf S. 18).

[LL20] S. Luber, N. Litzel. *Was sind Rohdaten?* 2020. URL: https://www.bigdata-insider.de/was-sind-rohdaten-a-920701/ (zitiert auf S. 5).

[Luc24] Lucidchart. *UML Aktivitätsdiagramm (Activity Diagram)*. 2024. URL: https://www.lucidchart.com/pages/de/uml-aktivitatsdiagramme (zitiert auf S. 40).

[Mat17] C. Mathis. „Data lakes“. In: *Datenbank-Spektrum* 17.3 (2017), S. 289–293 (zitiert auf S. 1).

[May24] N. Mayerhofer. *Constructing Knowledge Graphs From Unstructured Text Using LLMs*. 2024. URL: https://neo4j.com/developer-blog/construct-knowledge-graphs-unstructured-text/ (zitiert auf S. 64).

[MGS+19] M. N. Mami, D. Graux, S. Scerri, H. Jabeen, S. Auer, J. Lehmann. „Uniform access to multiform data lakes using semantic technologies“. In: *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*. 2019, S. 313–322 (zitiert auf S. 16).

[Mic23] Microsoft. *What is Azure Data Catalog?* 2023. URL: https://learn.microsoft.com/en-us/previous-versions/azure/data-catalog/overview (zitiert auf S. 13).

[Mic24a] Microsoft. *Data Catalog*. 2024. URL: https://azure.microsoft.com/de-de/products/data-catalog (zitiert auf S. 6, 7).

[Mic24b] Microsoft. *How to request access to a data product*. 2024. URL: https://learn.microsoft.com/en-us/purview/how-to-request-access-data-product (zitiert auf S. 7).

[Mic24c] Microsoft. *UInkrementelle Aktualisierung und Echtzeitdaten für Semantikmodelle*. 2024. URL: https://learn.microsoft.com/de-de/power-bi/connect-data/incremental-refresh-overview (zitiert auf S. 32).

[Mic24d] Microsoft. *What is the Microsoft Purview Data Catalog?* 2024. URL: https://learn.microsoft.com/en-us/purview/what-is-data-catalog (zitiert auf S. 13).

[Muc24] C. Mucci Tim; Stryker. *Unterschied zwischen DaaP und Datenprodukten*. 2024. URL: https://www.ibm.com/docs/de/data-product-hub/saas?topic=welcome-overview (zitiert auf S. 24, 26).

[Neo24a] Neo4j. *Neo4j AuraDB: Fully Managed Graph Database*. 2024. URL: https://neo4j.com/product/auradb/ (zitiert auf S. 43).

[Neo24b] Neo4j. *Northwind dataset*. 2024. URL: https://neo4j.com/docs/javascript-manual/current/ (zitiert auf S. 43).

[Neo24c] Neo4j. *Northwind dataset*. 2024. URL: https://github.com/neo4j-documentation/developer-resources/tree/gh-pages/data/northwind (zitiert auf S. 44, 46).

[Neo24d] Neo4j. *Tutorial: Import data from a relational database into Neo4j*. 2024. URL: https://neo4j.com/docs/getting-started/appendix/tutorials/guide-import-relational-and-etl/ (zitiert auf S. 44, 45).

[neo24a] neo4. *What is Cypher*. 2024. URL: https://neo4j.com/docs/getting-started/cypher/ (zitiert auf S. 43).

[neo24b] neo4j. *Neo4j Browser*. 2024. URL: https://neo4j.com/docs/browser-manual/current/ (zitiert auf S. 47).

[Nex24] Nexla. *Enterprise Data Catalog: Key Concepts Best Practices*. 2024. URL: https://nexla.com/data-fabric-architecture/enterprise-data-catalog/ (zitiert auf S. 7, 13).

[Nod24] Nodejs. *Introduction to Node.js*. 2024. URL: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs (zitiert auf S. 43).

[OAD+22] A. L. Opdahl, T. Al-Moslmi, D.-T. Dang-Nguyen, M. Gallofré Ocaña, B. Tessem, C. Veres. „Semantic knowledge graphs for the news: A review“. In: *ACM Computing Surveys* 55.7 (2022), S. 1–38 (zitiert auf S. 15).

[ODO24] B. Oliveira, A. Duarte, Ó. Oliveira. „Towards a Data Catalog for Data Analytics“. In: *Procedia Computer Science* 237 (2024), S. 691–700 (zitiert auf S. 6, 7).

[OO23] M. G. Ocaña, A. L. Opdahl. „A software reference architecture for journalistic knowledge platforms“. In: *Knowledge-Based Systems* 276 (2023), S. 110750 (zitiert auf S. 15).

[Par23] E. Parlament. *Massendaten: Definition, Vorteile, Herausforderungen (Infografik)*. 2023. URL: https://www.europarl.europa.eu/topics/de/article/20210211STO97614/massendaten-definition-vorteile-herausforderungen-infografik (zitiert auf S. 26).

[Par24] Parquet. *Apache Parquet*. 2024. URL: https://parquet.apache.org/ (zitiert auf S. 6).

[PUB23] D. Petrik, A. Untermann, H. Baars. „Functional Requirements for Enterprise Data Catalogs: A Systematic Literature Review“. In: *International Conference on Software Business*. Springer Nature Switzerland Cham. 2023, S. 3–18 (zitiert auf S. 1, 2, 6, 21, 23, 24).

[PVGW17] J. Z. Pan, G. Vetere, J. M. Gomez-Perez, H. Wu. *Exploiting linked data and knowledge graphs in large organisations*. Springer, 2017. ISBN: 978-3-319-45654-6. DOI: 10.1007/978-3-319-45654-6 (zitiert auf S. 10).

[PXNO23] C. Peng, F. Xia, M. Naseriparsa, F. Osborne. „Knowledge graphs: Opportunities and challenges“. In: *Artificial Intelligence Review* 56.11 (2023), S. 13071–13102 (zitiert auf S. 9, 10).

[Qli24] QlikTalend. *Managing email notifications*. 2024. URL: https://help.qlik.com/talend/en-US/data-catalog-administration-guide/8.0/managing-email-notifications (zitiert auf S. 7).

[Rag23] S. Raghavan. *Data Classification and Tagging: How to Marie Kondo Your Data Catalog and Spark Joy*. 2023. URL: https://atlan.com/data-classification-and-tagging/ (zitiert auf S. 7).

[Rea24] React. *React - The library for web and native user interfaces*. 2024. URL: https://react.dev/learn (zitiert auf S. 43).

[REF+19] C. Rudnik, T. Ehrhart, O. Ferret, D. Teyssou, R. Troncy, X. Tannier. „Searching news articles using an event knowledge graph leveraged by wikidata“. In: *Companion proceedings of the 2019 world wide web conference*. 2019, S. 1232–1239 (zitiert auf S. 15).

[Rig24] RightData. *Guide to Building a Data Catalog*. 2024. URL: https://www.getrightdata.com/resources/guide-to-building-a-data-catalog (zitiert auf S. 6, 7).

[RMXS19] N. F. Rajani, B. McCann, C. Xiong, R. Socher. „Explain Yourself! Leveraging Language Models for Commonsense Reasoning“. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Hrsg. von A. Korhonen, D. Traum, L. Màrquez. Florence, Italy: Association for Computational Linguistics, Juli 2019, S. 4932–4942. DOI: 10.18653/v1/P19-1487. URL: https://aclanthology.org/P19-1487/ (zitiert auf S. 35).

[RWE15] I. Robinson, J. Webber, E. Eifrem. *Graph databases: new opportunities for connected data*. Ö'Reilly Media, Inc.", 2015 (zitiert auf S. 26).

[SAP24] SAP. *Stammdaten*. 2024. URL: https://help.sap.com/docs/SAP_ERP/a58165ba587b4a7ebd5930c381a11649/3981cf535b804808e10000000a174cb4.html (zitiert auf S. 26).

[SAP25a] SAP. *Was ist Industrie 4.0?* 2025. URL: https://www.sap.com/germany/products/scm/industry-4-0/what-is-industry-4-0.html (zitiert auf S. 15).

[SAP25b] SAP. *What is Big Data?* 2025. URL: https://www.sap.com/products/technology-platform/what-is-big-data.html#:~:text=Big%20Data%20is%20defined%20as,re%20able%20to%20monitor%20digitally. (zitiert auf S. 40).

[Sca24] A. Scapicchio Mark; Forrest. *What is SSO?* 2024. URL: https://www.ibm.com/think/topics/single-sign-on#:~:text=Single%20sign%2Don%2C%20or%20SSO,when%20properly%20implemented%2C%20improves%20security. (zitiert auf S. 54).

[SD21] P. Sawadogo, J. Darmont. „On data lake architectures and metadata management". In: *Journal of Intelligent Information Systems* 56.1 (2021), S. 97–120 (zitiert auf S. 16).

[SGL+24] J. Schneider, C. Gröger, A. Lutsch, H. Schwarz, B. Mitschang. „The Lakehouse: State of the Art on Concepts and Technologies". In: *SN Computer Science* 5.5 (2024), S. 1–39 (zitiert auf S. 1, 4–6).

[SL22] J. Sequeda, O. Lassila. *Designing and building enterprise knowledge graphs*. Springer Nature, 2022 (zitiert auf S. 1, 10, 11, 17).

[STF+19] J. Strötgen, T.-K. Tran, A. Friedrich, D. Milchevski, F. Tomazic, A. Marusczyk, H. Adel, D. Stepanova, F. Hildebrand, E. Kharlamov. „Towards the bosch materials science knowledge base". In: (2019) (zitiert auf S. 16).

[VBJ20] W. Van Donge, N. Bharosa, M. Janssen. „Future government data strategies: data-driven enterprise or data steward? Exploring definitions and challenges for the government as data enterprise". In: *The 21st annual international conference on digital government research*. 2020, S. 196–204 (zitiert auf S. 19).

[vis24] vis.js. *Network*. 2024. URL: https://visjs.github.io/vis-network/docs/network/ (zitiert auf S. 43, 53).

[VZ14] A. Vaisman, E. Zimányi. „Data warehouse systems". In: *Data-Centric Systems and Applications* 9 (2014) (zitiert auf S. 4).

[WHC+19] X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua. „Kgat: Knowledge graph attention network for recommendation". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, S. 950–958 (zitiert auf S. 32).

[Won23] J. Wong. „Data Practices and Data Stewardship". In: *Interactions* 30.3 (2023), S. 60–63 (zitiert auf S. 19).

[Wut23] L. Wuttke. *Was macht ein Data Engineer? Überblick über das Berufsbild*. 2023. URL: https://datasolut.com/data-engineer-berufsbild/ (zitiert auf S. 19, 20).

[WW02] J. Webster, R. T. Watson. „Analyzing the past to prepare for the future: Writing a literature review". In: *MIS quarterly* (2002), S. xiii–xxiii (zitiert auf S. 14).

[WZXG18] H. Wang, F. Zhang, X. Xie, M. Guo. „DKN: Deep knowledge-aware network for news recommendation“. In: *Proceedings of the 2018 world wide web conference*. 2018, S. 1835–1844 (zitiert auf S. 32).

[YBA21] M. Yahya, J. G. Breslin, M. I. Ali. „Semantic web and knowledge graphs for industry 4.0“. In: *Applied Sciences* 11.11 (2021), S. 5110 (zitiert auf S. 15).

[YWC+18] J. Yan, C. Wang, W. Cheng, M. Gao, A. Zhou. „A retrospective of knowledge graphs“. In: *Frontiers of Computer Science* 12 (2018), S. 55–74 (zitiert auf S. 2, 9, 10).

[ZDED17] E. Zaidi, G. De Simoni, R. Edjlali, A. D. Duncan. „Data catalogs are the new black in data management and analytics“. In: *Gartner, Consultancy Report* (2017) (zitiert auf S. 6, 7).

[ZH10] X. Zhao, Z. Huang. „A quality evaluation approach for OLAP metadata of multidimensional OLAP data“. In: *2010 2nd IEEE International Conference on Information Management and Engineering*. IEEE. 2010, S. 357–361 (zitiert auf S. 4).

[ZLC+24] C. Zhang, Y. Li, S. Chen, S. Fan, W. Li. „Knowledge Graph Driven Recommendation System Algorithm.“ In: *CoRR* (2024) (zitiert auf S. 31).

[ZRKM20] J. Ziegler, P. Reimann, F. Keller, B. Mitschang. „A graph-based approach to manage CAE data in a data lake“. In: *Procedia CIRP* 93 (2020), S. 496–501 (zitiert auf S. 16).

[ZYST19] S. Zhang, L. Yao, A. Sun, Y. Tay. „Deep learning based recommender system: A survey and new perspectives“. In: *ACM computing surveys (CSUR)* 52.1 (2019), S. 1–38 (zitiert auf S. 32).

[ZZX+23] Y. Zheng, R. Zhang, J. Xu, Y. Gong, Y. You, Y. Zhang. „Research on the Data Governance in Power Industry Based on Data Owner“. In: *2023 Panda Forum on Power and Energy (PandaFPE)*. IEEE. 2023, S. 459–465 (zitiert auf S. 18).

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift