Strukturiertes Management von Forschungsdaten in den Ingenieurwissenschaften

Zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)
angenommene

DISSERTATION

von

M.Sc. Nico Brandt

Tag der mündlichen Prüfung:	2. Februar 2024
Hauptreferentin:	Prof. Dr. Britta Nestler
Korreferentin:	Prof. Dr. Bai-Xiang Xu

Kurzfassung

Die stetig wachsende Menge digitaler Daten aus verschiedenen Forschungsprozessen sowie das Potenzial, aus solchen Forschungsdaten mithilfe von Ansätzen aus den Datenwissenschaften neue Erkenntnisse zu gewinnen, machen deren strukturierte Verwaltung immer wichtiger. Dies ist die Aufgabe des sogenannten Forschungsdatenmanagements, das sämtliche zur systematischen Aufbereitung, Organisation und Bereitstellung von Forschungsdaten notwendigen Schritte umfasst, um die wissenschaftliche Aussagekraft der Daten zu erhalten. Zur Unterstützung eines strukturierten Forschungsdatenmanagements existieren unterschiedliche Ansätze, wie z. B. die Entwicklung und Etablierung von unterstützender Software, die auch als Forschungsdatenmanagementsoftware bezeichnet wird. Dieser Ansatz steht ebenfalls im Mittelpunkt der vorliegenden Arbeit und wird insbesondere im Kontext der Ingenieurwissenschaften betrachtet, die sich typischerweise durch eine hohe Heterogenität und Interdisziplinarität auszeichnen. Hierfür wird ein entsprechendes Systemkonzept entwickelt und implementiert, das neuartige Ansätze mit existierenden Technologien kombiniert. Durch den Einsatz innerhalb verschiedener Anwendungsfälle aus den Ingenieurwissenschaften wird die Funktionsweise und Flexibilität des entwickelten Systems aufgezeigt und evaluiert. Trotz potenzieller Herausforderungen bei der langfristigen und nachhaltigen Etablierung von Forschungsdatenmanagementsoftware zeigen die umgesetzten Ergebnisse das Potenzial für den Einsatz in einer breiten Spanne von Anwendungsfällen sowie für eine mögliche Adaption an weitere Forschungsdisziplinen.

Abstract

The constantly growing amount of digital data from various research processes and the potential to gain new insights from such research data with the aid of data science approaches make their structured management increasingly important. This is the task of the so-called research data management, which encompasses all steps necessary for the systematic preparation, organisation and provision of research data in order to preserve the scientific significance of the data. Various approaches exist to support structured research data management, such as the development and establishment of supporting software, also known as research data management software. This approach is also the focus of the present work and is considered in particular in the context of the engineering sciences, which are typically characterised by a high degree of heterogeneity and interdisciplinarity. For this purpose, a corresponding system concept is developed and implemented, which combines novel approaches with existing technologies. The functionality and flexibility of the developed system is demonstrated and evaluated through its use within various use cases from the engineering sciences. Despite potential challenges in the long-term and sustainable establishment of research data management software, the implemented results show the potential for use in a wide range of use cases and for possible adaptation to other research disciplines.

Vorwort und Danksagung

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Angewandte Materialien - Mikrostruktur-Modellierung und Simulation (IAM-MMS) des Karlsruher Instituts für Technologie (KIT). Ich danke dem Bundesministerium für Bildung und Forschung (BMBF) für die finanzielle Unterstützung während dieser Zeit.

Zuerst möchte ich mich bei Prof. Dr. Britta Nestler für die Betreuung während meiner Zeit als Doktorand bedanken und für die Möglichkeit, diese Arbeit in ihrem Institut mit vielen Freiheiten anfertigen zu dürfen. Ebenfalls danke ich Prof. Dr. Bai-Xiang Xu für die Bereitschaft, das Korreferat für diese Arbeit zu übernehmen.

Weiterhin bedanke ich mich bei meinem Gruppenleiter Dr. Michael Selzer für die intensive Betreuung und fachliche Unterstützung. Mein Dank gilt zudem meinen aktuellen bzw. ehemaligen Kollegen und Kolleginnen der „Kadi4Mat-Gruppe“, insbesondere Y. Zhao, E. Schoof, C. Herrmann, P. Zschumme, L. Griem, G. Tosato, M. Laqua, P. Altschuh, J. Grolig, A. Koeppe, J. Steinhülb, S. Kolli und M. Jayavarapu, jedoch ebenfalls einer Vielzahl nicht aufgelisteter Kollegen und Kolleginnen für zahlreiche Diskussionen, ein angenehmes und fruchtbares Arbeitsklima sowie abwechslungsreiche Gespräche während den Mittagspausen. Für die technische und administrative Unterstützung danke ich D. Frank, C. Ratz, H. Bayram, I. Heise, B. Hardt und M. Ritzinger.

Für die produktiven Kooperationen, die im Rahmen dieser Arbeit durchgeführt wurden, danke ich N. T. Garabedian, P. J. Schreiber, C. Greiner, B. Schmieg und J. Hubbuch. Mein weiterer Dank gilt allen Studenten, die als wissenschaftliche

Hilfskraft oder im Rahmen von Abschlussarbeiten mit mir zusammengearbeitet haben. Ebenfalls will ich den Entwicklern und Open-Source-Gemeinschaften sämtlicher Programmbibliotheken danken, welche bei der programmiertechnischen Umsetzung der Ergebnisse dieser Arbeit zum Einsatz kamen.

Schließlich möchte ich mich bei meiner Familie und insbesondere bei meinen Eltern bedanken, die mich in allen Lebenslagen uneingeschränkt unterstützen.

Karlsruhe, im November 2023 *Nico Brandt*

Inhaltsverzeichnis

1 Einleitung

Die strukturierte Verwaltung digitaler Daten spielt in nahezu sämtlichen Forschungsdisziplinen eine zunehmend wichtige Rolle [1, 2]. Dies ist nicht nur der stetig wachsenden Menge an Daten aus beispielsweise Experimenten oder Simulationen geschuldet [3], sondern ebenfalls dem Potenzial der Datenwissenschaften, möglichst automatisiert neues Wissen aus diesen Daten extrahieren zu können [4]. Letzteres wird manchmal auch als das vierte Paradigma der Wissenschaft bezeichnet [5]. Während die in einem solchen Kontext erzeugten Daten üblicherweise als Forschungsdaten bezeichnet werden, lässt sich deren Verwaltung unter dem Begriff Forschungsdatenmanagement (FDM) zusammenfassen. Die Bedeutung des FDMs prägt ebenfalls die Erwartungen von Förderorganisationen, was sich z. B. in den Leitlinien zur Sicherung guter wissenschaftlicher Praxis der Deutschen Forschungsgemeinschaft (DFG) [6] oder im von der Europäischen Union etablierten Forschungsprogramm Horizon Europe [7] widerspiegelt. Diesen Forderungen liegen häufig die sogenannten FAIR-Prinzipien [8] zugrunde, welche die Auffindbarkeit, Zugänglichkeit, Interoperabilität und Wiederverwendbarkeit von Forschungsdaten zum Ziel haben, um deren Nachhaltigkeit und wissenschaftliche Aussagekraft sicherzustellen. Dabei spielen vor allem Metadaten eine Rolle, eine besondere Form von Daten, die Informationen über unterschiedliche Eigenschaften anderer Daten enthalten.

Um die FAIR-Prinzipien vollständig umsetzen zu können, ist ein konsequent und strukturiertes FDM notwendig, angefangen mit der Planung eines Forschungsvorhabens über die eigentliche Erhebung und Aufbereitung der Forschungsdaten bis hin zu deren Publizierung oder Archivierung. Zur Unterstützung dieses Ablaufs

existieren unterschiedliche Arten von FDM-Software, wie z. B. elektronische Laborbücher, oft unter der englischen Bezeichnung Electronic Lab Notebook (ELN) zu finden, oder digitale Datenrepositorien. Diese können sowohl auf bestimmte Arbeitsabläufe oder Forschungsdisziplinen zugeschnitten sein, als auch generische Funktionalitäten zur Verfügung stellen. Für die Ingenieurwissenschaften, welche im Fokus dieser Arbeit stehen und sich durch eine hohe Heterogenität und Interdisziplinarität auszeichnen, ist insbesondere letztere Art von FDM-Software relevant. Neben der Implementierung von Software können ebenfalls weitere Aspekte wie z. B. die Etablierung von Datenkompetenzen (englisch: Data Literacy) oder die Konzeption von Metadatenstandards eine wichtige Rolle spielen, was u. a. im Rahmen von Initiativen wie NFDI4Ing [9] für den Bereich der Ingenieurwissenschaften verfolgt wird. Um einen Beitrag zur Unterstützung eines strukturierten FDMs in den Ingenieurwissenschaften zu leisten, lässt sich daher prinzipiell an mehreren Punkten anknüpfen. In dieser Arbeit liegt der Schwerpunkt auf der Konzeption und Implementierung einer praxisorientierten FDM-Software, die bestehende Systeme komplementiert, in bereits etablierte Arbeitsabläufe integriert werden kann und gleichzeitig die heterogenen Bedarfe der Ingenieurwissenschaften abdeckt. Die zentrale Forschungsfrage besteht in der Auseinandersetzung damit, welche Anforderungen eine entsprechende Software zur Erfüllung dieser Zielsetzungen umsetzen muss.

1.1 Gliederung

Die vorliegende Arbeit gliedert sich in insgesamt acht Kapitel. In diesem Kapitel wird die zentrale Forschungsfrage motiviert und die entsprechende Zielsetzung definiert, auf die sich der Rest der Arbeit stützt. In Kapitel 2 werden sämtliche für das Verständnis der Arbeit wichtigen Grundlagen erläutert. Anschließend werden in Kapitel 3 konkrete Initiativen, Architekturen und Technologien im Bereich des FDMs vorgestellt, an denen sich das anschließend in Kapitel 4 definierte Systemkonzept orientiert. Dessen programmiertechnische Umsetzung wird in Kapitel 5 erläutert und zusätzlich anhand eines einfachen Beispiels vorgestellt. In Kapitel 6

wird dagegen die Umsetzung konkreter Fallbeispiele erläutert, wobei drei aus den Ingenieurwissenschaften stammende Anwendungsfälle näher betrachtet werden. Eine abschließende Evaluation der Ergebnisse folgt in Kapitel 7, das ebenfalls einen Vergleich mit existierenden Systemen und eine Diskussion über langfristige Herausforderungen bei der Entwicklung und Etablierung von FDM-Software enthält. Zuletzt werden die zentralen Ergebnisse in Kapitel 8 zusammengefasst und ein Ausblick auf weiterführende Arbeiten und Ansätze gegeben.

2 Grundlagen

In diesem Kapitel werden sämtliche für das Verständnis der vorliegenden Arbeit notwendigen Grundlagen erläutert. Der Schwerpunkt liegt dabei auf den Definitionen verschiedener Begrifflichkeiten aus dem Bereich des FDMs sowie unterschiedlicher Arten von FDM-Software.

2.1 Forschungsdaten

Obwohl der Begriff der Forschungsdaten immer häufiger im Forschungsalltag verwendet wird, z. B. in den Leitlinien zur Sicherung guter wissenschaftlicher Praxis der DFG [6], existiert bisher keine allgemein gültige Definition. Die DFG selbst definiert den Begriff in deren Leitlinien zum Umgang mit Forschungsdaten als „Messdaten, Laborwerte, audiovisuelle Informationen, Texte, Surveydaten, Objekte aus Sammlungen oder Proben, die in der wissenschaftlichen Arbeit entstehen, entwickelt oder ausgewertet werden“ [10]. Jedoch können ebenfalls „[m]ethodische Testverfahren, wie Fragebögen, Software und Simulationen“ [10] als Forschungsdaten gewertet werden. Allgemeiner lässt sich die bereits im Jahr 2013 von Kindling und Schirmbacher verfasste Definition auffassen, nach welcher „alle digital vorliegenden Daten, die während des Forschungsprozesses entstehen oder ihr Ergebnis sind“ [11] als Forschungsdaten gelten. Bei keiner der beiden Definitionen wird explizit zwischen Primärdaten und Sekundärdaten unterschieden. Bei Primärdaten handelt es sich um sämtliche bzw. um alle aus einem Forschungsprozess gewonnenen Rohdaten, während Sekundärdaten lediglich aus den Primärdaten abgeleitet werden, z. B. durch die Anwendung eines Analysewerkzeugs.

Aufgrund der Vielfalt und Heterogenität der unterschiedlichen wissenschaftlichen Disziplinen, in denen Forschungsdaten entstehen und ausgewertet werden, ist es weder praktikabel noch sinnvoll, eine allgemeingültige Definition für Forschungsdaten aufzustellen. Dies trifft insbesondere auch auf die Ingenieurwissenschaften zu. Im weiteren Verlauf der Arbeit wird daher der Begriff der Forschungsdaten entsprechend der allgemeinen Definition von Kindling und Schirmbacher verwendet, aber zusätzlich um den Begriff der Metadaten erweitert, die im nächsten Abschnitt näher erläutert werden.

2.2 Metadaten

Bei Metadaten handelt es sich um eine besondere Form von Daten, die Informationen über andere Daten oder, allgemeiner, Ressourcen beinhalten, und damit auch um eine spezielle Form der Datendokumentation. Im Kontext digitaler Forschungsdaten werden Metadaten meist unabhängig von den eigentlichen Daten gespeichert, können jedoch auch mit diesen kombiniert werden. Ein Beispiel hierfür ist der Standard Extensible Metadata Platform (XMP) [12] zur Einbettung von Metadaten in Dateien verschiedener, gängiger Formate, wie z. B. PDF-, JPEG- oder MP3-Dateien. Im Forschungsumfeld wird ebenfalls häufig das Datenformat HDF5 [13] zum Speichern größerer Datenmengen eingesetzt. Dieses verwendet eine Struktur, die einem Dateiverzeichnis ähnelt, um prinzipiell beliebige Daten speichern und organisieren zu können. Durch die Verwendung von Attributen können die Daten optional mit zusätzlichen Metadaten versehen werden. Für separat gespeicherte Metadaten werden typischerweise textbasierte, strukturierte Datenformate wie JSON oder XML verwendet, die mithilfe von Containerformaten wie z. B. RO-Crate [14] oder BagIt [15] ebenfalls mit den entsprechenden Forschungsdaten in Form einfacher Archive gebündelt werden können.

Metadaten können hinsichtlich ihres Inhalts verschiedene Funktionen erfüllen. Grundsätzlich lässt sich zwischen administrativen bzw. technischen und fachlichen Metadaten unterscheiden. Erstere umfassen allgemeine Metadaten, wie z. B. Titel, Autor, diverse Zeitstempel oder die Größen und Formate digitaler

Forschungsdaten, unabhängig von der jeweiligen Forschungsdisziplin. Fachliche Metadaten enthalten dagegen disziplinspezifische Informationen zum Inhalt und der Entstehung der Daten und finden daher teilweise auch selbst als Forschungsdaten Verwendung. Eine genaue Definition der Metadaten und ihres Status ist daher schwierig, genau wie bei den Forschungsdaten selbst. Da Metadaten jedoch einen essenziellen Bestandteil von Forschungsdaten darstellen [11], werden insbesondere die fachlichen Metadaten im weiteren Verlauf der Arbeit ebenfalls als Teil der Forschungsdaten betrachtet. Diese Auffassung ist auch zur Beschreibung von Forschungsprozessen oder -objekten hilfreich, denen selbst keine Forschungsdaten zugrunde liegen, z. B. die Verarbeitung einer physischen Probe mithilfe eines Laborgeräts oder das Laborgerät selbst [8]. In diesem Fall können unter Umständen lediglich die Metadaten einen Aufschluss über die Umgebungsbedingungen des Prozesses geben, selbst wenn dieser nur einen Zwischenschritt innerhalb eines Gesamtprozesses darstellt.

2.2.1 Metadatenschemata

Im Gegensatz zu textuellen Datendokumentationen, die üblicherweise in unstrukturierter Form vorliegen und von Menschen gelesen und interpretiert werden, ist die Lesbarkeit und Interpretation durch Maschinen ein zentrales Merkmal von Metadaten. Während eine grundlegende Maschinenlesbarkeit durch die Verwendung standardisierter und strukturierter Datenformate wie JSON und XML einfach sichergestellt werden kann, hängt die Interpretation der Metadaten stark vom jeweiligen Forschungskontext ab. Um aus verschiedenen Forschungsprozessen gewonnene Metadaten vergleichen zu können, bieten sich daher standardisierte Metadatenschemata an, die den Metadaten eine definierte Struktur und Semantik zuweisen können. Wie bei den Metadaten selbst, wird auch hier zwischen administrativen, oft multidisziplinären, und fachspezifischen Schemata unterschieden, die sich je nach Anwendungsfall und Forschungsdisziplin für verschiedene Zwecke eignen können. Eine Übersicht über beiden Arten von standardisierten Metadatenschemata liefert z. B. der von der Research Data Alliance (RDA) entwickelte

Metadata Standards Catalog [16]. Eine Vorstellung ausgewählter Schemata findet sich in Kapitel 3.

2.3 Ontologien

Während durch Metadatenschemata eine einheitliche Struktur für bestimmte Arten von Metadaten vorgegeben werden kann, und damit bereits ein hohes Maß an Interoperabilität auf syntaktischer Ebene möglich ist, ist die semantische Interoperabilität nicht immer vollständig gewährleistet. Für diese Zwecke können zusätzlich Ontologien zum Einsatz kommen [17]. Das Konzept der Ontologien stammt ursprünglich aus der Philosophie, wobei eine Ontologie eine systematische Beschreibung der „Existenz“ darstellt [18]. Ausgehend vom Bereich der künstlichen Intelligenz [19] hat sich der Begriff allmählich zu der modernen, in den Informationswissenschaften gebräuchlichen Definition gewandelt: Eine formale Beschreibung von Konzepten und deren Beziehungen untereinander innerhalb einer oder mehrerer Domänen. Das Ziel von Ontologien ist die Formalisierung von Wissen, das sowohl Teil der Ontologien selbst ist als auch von diesen abgeleitet werden kann, womit Ontologien über die Möglichkeiten anderer semantischer Modelle wie Glossare, Taxonomien und Thesauri hinausgehen [20]. Einen Einsatz finden Ontologien z. B. als Teil des Semantic Webs [21], eine Erweiterung des World Wide Webs, um dessen Inhalte maschinenlesbar und -interpretierbar zu machen.

Ontologien können auf verschiedene Weise spezifiziert werden und unterscheiden sich daher in den verwendeten Komponenten oder zumindest in deren Bezeichnung. Häufig verwendete Grundkomponenten sind Klassen, Instanzen, Eigenschaften und Relationen. Instanzen beschreiben konkrete Individuen, z. B. eine bestimmte Person, die jeweils Teil einer Klasse und damit eines bestimmten Typs sind. Klassen und Instanzen können sowohl Eigenschaften besitzen, wie z. B. den Namen einer Person, sowie in bestimmten Relationen zueinanderstehen, z. B. in Form unterschiedlicher Verwandtschaftsverhältnisse.

Ein weit verbreiteter Standard zur Definition von Ontologien, insbesondere für den Einsatz innerhalb des Semantic Webs, ist die Web Ontology Language (OWL) [22], die wiederum auf dem Resource Description Framework (RDF) [23] aufbaut. RDF beschreibt Metadaten in Form von gerichteten Graphen, die jeweils aus Tripeln aufgebaut sind. Jedes Tripel besteht aus einem Subjekt (z. B. *Karlsruhe*), einem Prädikat (z. B. *ist*), und einem Objekt (z. B. *Stadt*), womit beliebige Ressourcen und deren Relationen zueinander beschrieben werden können. Jede Komponente eines Tripels muss mithilfe eines Internationalized Resource Identifiers (IRI) spezifiziert werden, eine internationalisierte Form der Uniform Resource Identifier (URI), lediglich Objekte dürfen auch aus einem einfachen Literal (z. B. textuelle Werte oder Zahlen) bestehen. Zur Serialisierung sind verschiedene Formate möglich wie z. B. JSON-LD [24] oder Turtle [25], während spezielle Formen von Datenbanken, sogenannte Triplestores, die Persistierung und Abfrage von Tripeln mithilfe der Abfragesprache SPARQL (SPARQL Protocol And RDF Query Language) ermöglichen. Um die volle Komplexität von Ontologien erfassen zu können, erweitert OWL den Funktionsumfang von RDF um zusätzliche Sprachkonstrukte. Eine kurze Vorstellung ausgewählter Ontologien und verwandter Technologien findet sich in Kapitel 3.

2.4 Forschungsdatenmanagement

Unter FDM werden sämtliche Schritte verstanden, die zur systematischen Erstellung bzw. Aufbereitung, Organisation, Speicherung und Bereitstellung von Forschungsdaten notwendig sind [26–28]. Ziel des FDMs ist es, den gemeinsamen und langfristigen Zugriff auf Forschungsdaten zu ermöglichen und deren wissenschaftliche Aussagekraft zu erhalten, unabhängig davon, von wem die Daten ursprünglich erzeugt oder ausgewertet wurden [26].

Zum besseren Verständnis, welche Aspekte eines Forschungsprozesses durch ein strukturiertes FDM unterstützt werden können, ist der sogenannte Forschungsdatenlebenszyklus hilfreich. Dieser wurde vor allem durch das vom Digital Curation Centre (DCC) entwickelte Curation Lifecycle Model [29] populär, es existieren

jedoch eine Vielzahl unterschiedlicher Definitionen und Interpretationen [30]. Der Forschungsdatenlebenszyklus bildet die verschiedenen, möglichen Aufgabenbereiche des FDMs auf die jeweils dazu passenden Phasen innerhalb eines Forschungsprozesses ab, angefangen von der Planung bis zur Wiederverwendung von Forschungsdaten. Zwar stellen Forschungsdatenlebenszyklen typischerweise einen idealistischen und im Vergleich zur Praxis zu linearen Ablauf dar [29], dennoch eignen sich diese zur Veranschaulichung der wesentlichen Bestandteile des FDMs und zur Orientierung für das Planen von Forschungsvorhaben [28].

Ein Modell eines typischen Forschungsdatenlebenszyklus ist in Abbildung 2.1 dargestellt. Dieser ist in die folgenden sechs Phasen unterteilt:

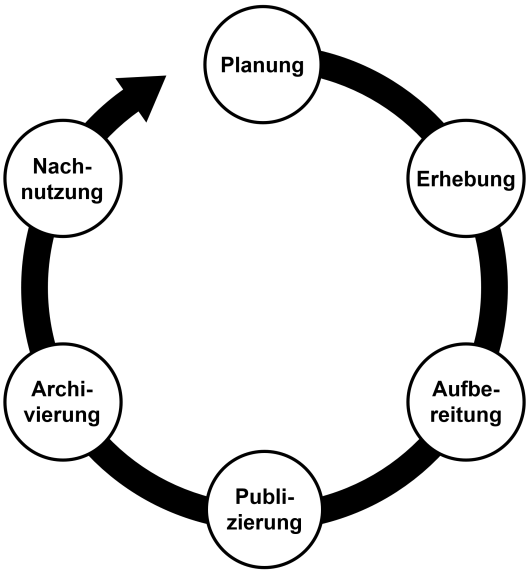

Abbildung 2.1: Darstellung eines typischen Forschungsdatenlebenszyklus, unterteilt in sechs Phasen.

- Planung: Die meisten Forschungsdatenlebenszyklen beginnen mit der Planung eines Forschungsvorhabens im Hinblick der auf die zur Beantwortung einer Forschungsfrage generierten Daten. Immer häufiger kommen dabei sogenannte Datenmanagementpläne (DMPs) [26, 31] zum Einsatz. Ziel solcher Pläne ist es, die übrigen Stadien des Forschungsdatenlebenszyklus bereits im Voraus zu koordinieren, angefangen von der Nachnutzung existierender Daten über die Beschreibung, Speicherung und Publizierung der gesammelten und generierten Daten bis hin zu der erforderlichen Software, Hardware und entsprechendem Personal. Letzteres spielt für Förderorganisationen eine zunehmend wichtige Rolle, weshalb entsprechende Pläne z. B. im Forschungsprogramm Horizon Europe explizit gefordert werden [7].

- Erhebung: Diese Phase stellt die Generierung der Rohdaten bzw. Primärdaten und entsprechender Metadaten innerhalb eines Forschungsvorhabens dar, was durch Experimente, Simulationen, Messungen oder andere Forschungsprozesse erfolgen kann.

- Aufbereitung: Die Aufbereitung der Daten beinhaltet sowohl deren Analyse und Interpretation, und damit die Generierung von Sekundärdaten und entsprechenden, weiteren Metadaten, als auch die Sicherung der Datenqualität durch die Bereinigung und Validierung aller Daten. Insbesondere bei personenbezogenen Daten kann auch eine Anonymisierung oder Pseudonymisierung der Daten notwendig sein.

- Publizierung: Die aufbereiteten Daten können anschließend, möglichst zugänglich, publiziert werden, inklusive aller zum Verständnis der Daten relevanten Metadaten und Beschreibungen. Die Nachnutzung der Daten kann durch die Angabe einer Lizenz und optionaler Embargofristen gesteuert werden.

- Archivierung: Zusätzlich zur Publizierung werden alle langfristig relevanten Daten auf geeigneten Medien archiviert. Neben den Metadaten ist insbesondere die Nutzung von standardisierten und in Zukunft weiterhin nutzbaren Datenformaten essenziell. Im Gegensatz zu Backups liegt der Schwerpunkt

bei der Archivierung lediglich auf ausgewählten Daten und nicht auf der reinen Prävention von Datenverlust, die bereits in der Erhebungsphase wichtig ist.

- Nachnutzung: Wurden die Daten strukturiert erfasst, passend aufbereitet und publiziert, ist deren Nachnutzung für weitere Forschungsvorhaben möglich. Dementsprechend können solche Daten wieder in der Planungsphase berücksichtigt werden, womit sich der Forschungsdatenlebenszyklus schließt.

2.5 Die FAIR-Prinzipien

Ein im Kontext des FDMs mittlerweile sehr häufig verwendeter Begriff sind die sogenannten FAIR-Prinzipien [8]. Bei diesen handelt es sich um vier Leitlinien, deren Umsetzung zu einem nachhaltigeren FDM beitragen soll und dabei insbesondere Wert auf die Maschinenlesbarkeit von publizierten Daten und Metadaten legt. Die Abkürzung FAIR steht für Findable (deutsch: auffindbar), Accessible (deutsch: zugänglich), Interoperable (deutsch: interoperabel) und Reusable (deutsch: wiederverwendbar). Zusammengefasst lassen sich die Leitlinien wie folgt beschreiben:

- Findable: Daten sollen mit aussagekräftigen Metadaten verknüpft werden. Weiterhin sollen sowohl Daten als auch Metadaten mit global eindeutigen und persistenten Identifikatoren ausgestattet werden und in einer durchsuchbaren Form indexiert werden.
- Accessible: Daten und Metadaten sollen über deren Identifikatoren mithilfe von standardisierten und offenen Protokollen abrufbar sein. Sollten Daten in Zukunft nicht mehr verfügbar sein, sollen es zumindest die entsprechenden Metadaten noch sein.
- Interoperable: Daten und Metadaten verwenden eine formale, zugängliche und möglichst allgemein anwendbare Sprache zur Repräsentation von

Wissen. Verwandte Daten und Metadaten sollen zudem durch qualifizierte Verweise referenziert werden.

- Reusable: Daten und Metadaten sollen durch relevante Attribute so beschrieben werden, dass deren Nachnutzung gewährleistet ist. Dies beinhaltet die Nutzung von etablierten Standards, die Beschreibung der Datenherkunft (englisch: Data Provenance) und das Festlegen von Nutzungslizenzen.

Während vergleichbare Publikationen meist auf spezifische Anwendungsgebiete oder Domänen abzielen [32, 33], handelt es sich bei den FAIR-Prinzipien um sehr allgemein gehaltene und domänenunabhängige Leitlinien, die selbst keine Vorgaben für ihre konkrete Umsetzung beinhalten [34, 35]. Um die konkrete Umsetzung der Leitlinien zu unterstützen, sind im Laufe der Zeit unterschiedliche Erweiterungen entstanden, wie z. B. die Anwendung der FAIR-Prinzipien auf Forschungssoftware [36] oder die FAIRsFAIR Data Object Assessment Metrics [37] zur Evaluierung konkreter Datensätze. Aufgrund ihrer weiten Verbreitung werden die FAIR-Prinzipien parallel zu DMPs auch in den Anforderungen von Förderorganisationen für das FDM berücksichtigt, z. B. als Teil der Leitlinien zur Sicherung guter wissenschaftlicher Praxis der DFG [6].

2.6 Forschungsdatenmanagementsoftware

Bei FDM-Software handelt es sich um unterschiedliche Arten von Systemen oder Softwarewerkzeugen, deren allgemeines Ziel es ist, Forscher bei der Umsetzung eines strukturierten FDMs zu unterstützen. Diese Art von Software kann als Teil einer sogenannten Forschungsdateninfrastruktur bereitgestellt werden und lässt sich teilweise von „regulärer“ Forschungssoftware abgrenzen, bei der primär die Generierung oder Analyse von Forschungsdaten im Fokus steht [38]. Dennoch kann FDM-Software auch selbst z. B. Analysewerkzeuge beinhalten oder zumindest deren Verwendung vereinfachen, weshalb eine direkte Abgrenzung der beiden Softwaretypen nicht immer möglich ist. Im weiteren Verlauf der Arbeit wird daher sämtliche Software, welche die Umsetzung einer oder mehrerer Phasen

des Forschungsdatenlebenszyklus mit Fokus auf dem eigentlichen Management der Forschungsdaten unterstützt, als FDM-Software bezeichnet. Die Entwicklung von FDM-Software wird teilweise auch dem allgemeinen Begriff des Research Software Engineerings untergeordnet [39]. Im Folgenden werden die wichtigsten Arten von FDM-Software erläutert, sowie deren mögliche Verwendung innerhalb des Forschungsdatenlebenszyklus. Ausgewählte Systeme werden in Kapitel 3 näher vorgestellt.

2.6.1 Datenmanagementplan-Software

Zur Erstellung von DMPs innerhalb der Planungsphase von Forschungsvorhaben lassen sich klassischerweise entsprechende Checklisten oder Leitlinien [40, 41] verwenden, jedoch sind im Laufe der Zeit unterschiedliche Softwarewerkzeuge entstanden, um bei der Konzeption solcher Pläne zu unterstützen [42–44]. Neben mehr Flexibilität im Vergleich zu statischen Checklisten können z. B. Templates angeboten werden, um die Anforderungen verschiedener Förderorganisationen abbilden zu können. Auch die Wiederverwendbarkeit von DMPs wird durch die Nutzung entsprechender Systeme vereinfacht.

2.6.2 Elektronische Laborbücher

ELNs spielen eine zunehmend wichtige Rolle innerhalb von Forschungsdateninfrastrukturen [45, 46]. Bei diesen handelt es sich um Software zur Erfassung und Dokumentation von Forschungsprozessen und dadurch entstehender Forschungsdaten, entweder als Ersatz oder parallel zur Verwendung klassischer, papierbasierter Laborbücher. Aus Sicht des in Abbildung 2.1 dargestellten Forschungsdatenlebenszyklus können ELNs insbesondere bei der Erhebung und Aufbereitung eine Rolle spielen.

ELNs lassen sich in verschiedene Kategorien aufteilen, wobei eine zentrale Rolle spielt, ob ein System fachspezifische oder generische Funktionalitäten anbietet [47, 48]. Üblicherweise werden ELNs innerhalb eines eingeschränkten Nutzerkreises

verwendet, z. B. von bestimmten Arbeitsgruppen eines Instituts, können prinzipiell jedoch auf unterschiedlichen Ebenen eingesetzt werden. ELNs kommen insbesondere bei wissenschaftlichen Experimenten zum Einsatz, wie sie z. B. in der Chemie oder den Lebenswissenschaften durchgeführt werden. Gleichzeitig geht das Verständnis von ELNs zunehmend über den einfachen Ersatz von klassischen Laborbüchern hinaus, und damit auch deren Verwendung in anderen Forschungsdisziplinen. Insbesondere verschwimmt die Grenze zwischen unstrukturierten und strukturierten Daten, wobei letztere eher in Repositorien oder, im Kontext von experimentellen Forschungsarbeiten, in Labor-Informations- und Management-Systemen (LIMS) zu finden sind [49, 50]. Auch der Aspekt der Laborautomatisierung und die automatisierte Datenakquise von Laborgeräten kann daher in den Aufgabenbereich von ELNs fallen [51].

2.6.3 Repositorien

Als Repositorien wurden ursprünglich Lagerorte zur geordneten Aufbewahrung von physischen Dokumenten bezeichnet, die entweder öffentlich oder einem eingeschränkten Nutzerkreis zur Verfügung stehen, wobei es sich bei diesen im Gegensatz zu Archiven nicht um überwiegend historische Dokumente handelt. Das moderne, und in dieser Arbeit verwendete, Verständnis von Repositorien beinhaltet dagegen die strukturierte Speicherung digitaler Objekte mithilfe entsprechender, meist webbasierter, Software, wofür ebenfalls der englische Begriff Repository gebräuchlich ist. Repositorien stellen eine der wichtigsten Softwarekomponenten einer Forschungsdateninfrastruktur dar [52, 53] und spielen insbesondere in den letzten drei Phasen (Publizierung, Archivierung und Nachnutzung) des in Abbildung 2.1 dargestellten Forschungsdatenlebenszyklus eine Rolle.

Grundsätzlich lassen sich die im Kontext des FDMs eingesetzten Repositorien in vielerlei Hinsicht differenzieren und werden daher für sehr unterschiedliche Zwecke genutzt [54, 55]. Eines der wichtigsten Unterscheidungsmerkmale ist, ähnlich zu ELNs, der Fokus auf entweder fachspezifische oder generische Forschungsdaten und Metadaten, wobei prinzipiell auch Kombinationen möglich sind. Weiterhin

können die gespeicherten Forschungsdaten sowohl internen Zwecken dienen, z. B. dem strukturierten FDM und Datenaustausch innerhalb einzelner Institute, als auch Publikationen darstellen. Letztere werden üblicherweise mit einem persistenten Identifikator (englisch: Persistent Identifier, kurz PID) ausgestattet, z. B. einem Digital Object Identifier (DOI), und können sowohl alleinstehen als auch der Ergänzung klassischer Textpublikationen dienen. Schließlich kann zwischen institutionellen und zentral bereitgestellten Repositorien unterschieden werden, wobei im Prinzip beide Arten von Repositorien für die oben erläuterten Zwecke eingesetzt werden können.

Speziell zur Evaluierung von Repositorien sind im Laufe der Zeit unterschiedliche Metriken entstanden, z. B. die an die FAIR-Prinzipien angelehnten TRUST-Prinzipien [56], welche u. a. die Sicherheit, Zuverlässigkeit und Nachhaltigkeit von Repositorien bewerten. Auch entsprechende Zertifikate werden vergeben, wie z. B. das CoreTrustSeal [57], das insbesondere für öffentlich zugängliche Repositorien mit Schwerpunkt auf der Publizierung von Forschungsdaten relevant ist.

2.6.4 Workflow-Management-Systeme

Bei Workflow-Management-Systemen handelt es sich allgemein betrachtet um Softwarewerkzeuge zur Modellierung und Ausführung von Arbeitsabläufen, die üblicherweise als Workflow bezeichnet werden. Workflows und Workflow-Management-Systeme sind bereits seit den 1990er-Jahren ein Begriff, weshalb es eine Vielzahl von Vorstellungen über ihren Inhalt und Aufbau gibt [58]. In dieser Arbeit werden Workflows als ein allgemeines Konzept betrachtet, das mithilfe einer formalen Beschreibung eine wohldefinierte Abfolge von sequentiellen oder parallelen Schritten darstellt. Zu diesen Schritten lassen sich hauptsächlich Softwarewerkzeuge zählen, deren Ausführung in der Regel automatisiert werden kann, z. B. die Anwendung eines Analysewerkzeugs auf einen Datensatz oder die Steuerung und anschließende Datenakquise eines Laborgeräts über entsprechende Softwareschnittstellen. Ein Workflow-Management-System ist für die Verwaltung

und eigentliche Ausführung der Workflows zuständig, kann jedoch auch deren Erstellung unterstützen, z. B. mithilfe einer grafischen Benutzeroberfläche (englisch: Graphical User Interface, kurz GUI).

Nach dieser Definition kann der Schwerpunkt von Workflow-Management-Systemen überwiegend auf computergestützte Wissenschaftsbereiche gelegt werden und hier zur strukturierten Spezifikation der Datenherkunft beitragen [59]. Dennoch bestehen etliche Parallelen zum zuvor bereits beschriebenen, modernen Verständnis von ELNs, insbesondere was den Bereich der Laborautomatisierung betrifft. Aufgrund ihrer Allgemeinheit können Workflow-Management-Systeme in nahezu allen Phasen des in Abbildung 2.1 dargestellten Forschungsdatenlebenszyklus zum Einsatz kommen.

2.6.5 Registries und Terminologieservices

Registries umfassen eine Gruppe von Systemen, die hauptsächlich der Referenzierung unterschiedlicher Konzepte dienen, wie z. B. Datentypen, Metadatenschemata oder sogar Autoren [60]. Im Kontext der vorliegenden Arbeit sind insbesondere Terminologieservices relevant, eine spezielle Art von Registry zur Ablage und Bereitstellung von Konzepten, deren Beziehungen und entsprechende Terminologien um diese repräsentieren zu können [61]. Die Inhalte solcher Services sind in deren Aufbau oft eng mit den Komponenten von Ontologien verwandt, z. B. durch die Verwendung von Klassen, konkreter Instanzen und Relationen, jedoch können auch die zugrunde liegenden, vollständigen Ontologien hinterlegt werden. Ziel der Terminologieservices ist es, individuelle Terme auffindbar und über einen eindeutigen Identifikator, üblicherweise ein IRI oder URI, referenzierbar zu machen. Dieser kann wiederum z. B. innerhalb von Metadatenschemata verwendet werden, um bei der Standardisierung der Metadatenelemente zu unterstützen, insbesondere im Hinblick auf die semantische Interoperabilität. Ähnlich wie andere Arten von FDM-Software können Terminologieservices fachspezifisch sein oder Konzepte und Ontologien aus unterschiedlichen Domänen beinhalten.

3 Stand der Forschung

In diesem Kapitel werden aktuelle Initiativen und konkrete Technologien im Bereich des FDMs vorgestellt, wobei der Schwerpunkt auf Entwicklungen liegt, welche für die Konzeption, Umsetzung und Evaluierung der Ergebnisse dieser Arbeit am relevantesten sind. Die beschriebenen Entwicklungen lassen sich teilweise dem Bereich der Ingenieurwissenschaften zuordnen, finden größtenteils jedoch auch allgemein Anwendung.

3.1 Initiativen und Data Stewardship

Strukturiertes FDM kann prinzipiell auf unterschiedlichen Ebenen erfolgen, z. B. international, national oder institutionell. Auf internationaler bzw. europäischer Ebene spielt insbesondere die European Open Science Cloud (EOSC) [62] eine Rolle. Bei der EOSC handelt es sich um eine föderierte Forschungsdateninfrastruktur, die seit Ende des Jahres 2018 von der Europäischen Kommission gefördert wird. Hierbei steht weniger die Entwicklung neuer Systeme im Vordergrund, sondern der Zusammenschluss bereits existierender Technologien und Infrastrukturen, wobei von Repositorien über ELNs bis hin zu Analysewerkzeugen eine breite Auswahl an Software zum Einsatz kommen kann [63]. Die eigentlichen Forschungsdaten können dabei unterschiedlichen Disziplinen und damit auch den Ingenieurwissenschaften zugeordnet werden, der Fokus liegt jedoch auf einem disziplinübergreifenden Austausch.

Anders verhält es sich bei der innerhalb von Deutschland etablierten und durch den Bund und die Länder finanzierten Nationalen Forschungsdateninfrastruktur

(NFDI) [64]. Diese setzt sich aus unterschiedlichen Konsortien zusammen, um die Entwicklung fachspezifischer Lösungen einzelner Wissenschaftsbereiche voranzutreiben. Für den Bereich der Ingenieurwissenschaften ist z. B. das Konsortium NFDI4Ing [9] zuständig, das bereits seit Ende des Jahres 2020 gefördert wird. Die Zielsetzung von NFDI4Ing ist es, die große Vielfalt an technischen Ansätzen des FDMs in den Ingenieurwissenschaften in einer begrenzten Anzahl von gemeinsamen Standards und Zielen zu konsolidieren [65]. Dazu gehört neben unterschiedlicher Forschungssoftware auch die Etablierung von Datenkompetenzen. Die NFDI als Ganzes ist ebenfalls Mitgliedsorganisation in der EOSC [63], wodurch die internationale Zusammenarbeit mit ähnlichen Vorhaben verstärkt werden soll.

Trotz solcher Vorhaben ist im Laufe der Zeit bereits eine heterogene Landschaft an landesweiten oder institutionellen Initiativen, Richtlinien und Lösungen entstanden [66–69]. Auf institutioneller Ebene sind insbesondere Bibliotheken zunehmend in die Prozesse des FDMs an ihren Universitäten eingebunden [70, 71]. Das bereits vorhandene Verständnis von Bibliothekaren für eine strukturierte Organisation von Informationen, unabhängig vom konkreten Informationsmedium, lässt sich prinzipiell auch auf das Management von Forschungsdaten ausweiten. Dies gilt insbesondere für die Expertise bei der Verwaltung von Metadaten. Die Rollen, die Bibliothekare dabei annehmen können, beinhalten das Engagement für das Teilen und die öffentliche Bereitstellung von Forschungsdaten, die Lehre von Datenkompetenzen oder die Unterstützung bei der Kuration und Qualitätskontrolle institutioneller Repositorien [72]. Die angebotenen Dienstleistungen fokussieren sich somit vor allem auf Beratungs- und Betreuungsdienste und weniger auf technische Dienstleistungen, deren Umsetzung oft zusätzliches Fachwissen voraussetzt. Dies wiederum legt eine Zusammenarbeit mit weiteren Akteuren nahe, wie z. B. universitäre Rechenzentren. Je nach Fachgebiet müssen ebenfalls Rechtsabteilungen sowie die wissenschaftlichen Mitarbeiter unterschiedlicher Disziplinen selbst in die Entwicklung unterstützender Prozesse mit einbezogen werden [66].

Diese Faktoren können es für Bibliothekare schwer machen, sich als Hauptakteure für das FDM zu positionieren. Aus diesen Anforderungen heraus hat sich in den

letzten Jahren vermehrt die Begrifflichkeit des Data Stewardships etabliert, das sich gezielt mit der Unterstützung aller Phasen des Forschungsdatenlebenszyklus befasst. In einigen Disziplinen, wie z. B. den biomedizinischen Wissenschaften, können ebenfalls Aspekte wie Datenschutz in den Bereich des Data Stewardships fallen [73]. Die entsprechende Rolle des Data Stewards wird zunehmend in Forschungsprojekte eingegliedert, wie z. B. auch im Exzellenzcluster POLiS (Post Lithium Storage) [74] des Karlsruher Instituts für Technologie (KIT) und der Universität Ulm. Auch innerhalb des EOSC werden die Tätigkeiten, Verantwortlichkeiten und Karrierewege von Data Stewards evaluiert, was auch die mögliche Einbettung entsprechender Kompetenzen in Lehrpläne umfasst [75].

Zusammenfassend lässt sich FDM als eine Aufgabe unterschiedlichster Akteure sehen, deren Aufgaben je nach Position und Fachgebiet von Beratungsdiensten bis hin zur Entwicklung unterstützender und anwendungsspezifischer Werkzeuge reichen können. FDM kann ebenfalls nicht als einheitlicher Satz von Diensten betrachtet werden, der in verschiedenen wissenschaftlichen Einrichtungen lediglich dupliziert werden muss, sondern es handelt sich um individuelle Lösungen, die unter Berücksichtigung unterschiedlicher Faktoren entwickelt werden [76]. Dies ist insbesondere im Kontext der Ingenieurwissenschaften relevant, die ein weites Spektrum sehr unterschiedlicher Methoden und damit Daten abdecken. Daher wird tendenziell die Verantwortung für die Bewältigung dieser Herausforderungen langfristig weiterhin bei den individuellen Institutionen oder Arbeitsgruppen liegen [70], kann jedoch von zentralisierten Entitäten in vielerlei Hinsicht unterstützt werden.

3.2 Architekturen für das Forschungsdatenmanagement

Da die FAIR-Prinzipien viel Interpretationsfreiraum für deren technische Umsetzung bieten, existieren etliche, interdisziplinäre Architekturstandards in unterschiedlichen Reifegraden zur Verwaltung FAIRer Forschungsdaten [77]. Ein

bereits im Kontext der EOSC vorgeschlagenes und diskutiertes Konzept stellen die sogenannten FAIR Digital Objects (FDOs) dar [78, 79]. Im Kontext der FDOs handelt es sich bei einem regulären digitalen Objekt um ein Forschungsdatum, das durch einen PID referenziert und durch entsprechende Metadaten beschrieben wird, die selber wiederum ein solches digitales Objekt darstellen. Durch die Verwendung offener und standardisierter Datenformate und möglichst auf etablierten Schemata oder Terminologien aufbauende Metadaten können diese Objekte zunehmend FAIR werden, was den Fokus von FDOs darstellt. Die Implementierung von FDOs kann unterschiedliche Dienste umfassen, insbesondere Repositorien für den zentralen Zugriff auf Daten und Metadaten sowie Registries. Da es sich hierbei um ein sehr allgemeines Konzept handelt, existieren zudem verschiedene darauf aufbauende Arbeiten. Ein Beispiel ist das FDO Framework [80], welches die Wiederverwendbarkeit und Interoperabilität von FDOs durch eine standardisierte Organisation und einheitliche Zugriffsprotokolle verbessern soll. Parallel existieren bereits konkrete Ansätze zur Nutzung von FDOs innerhalb bestimmter Domänen in Kombination mit Ontologien und anderen Metadatenstandards [81].

Ein weiterer, nennenswerter Standard ist die Linked Data Platform (LDP) [82], die sich im Vergleich zu Ansätzen wie dem FDO Framework durch eine geringere Komplexität auszeichnet. Ein LDP unterstützender Server muss lediglich das Protokoll HTTP (Hypertext Transfer Protocol) unterstützen, um entsprechende LDP-Ressourcen erstellen oder abrufen zu können. Bei diesen wird zwischen RDF-basierten und sonstigen Ressourcen unterschieden, wobei es sich bei letzteren um reguläre Binärdaten unterschiedlicher Formate handelt. Beide Ressourcentypen können ebenfalls in, optional geschachtelten, Containern gruppiert werden, um deren Organisation zu erleichtern. Obwohl LDP diverse Aspekte offenlässt, wie z. B. die Verwendung von PIDs, kann der Standard aufgrund seiner Einfachheit als eine gemeinsame Grundlage für andere Architekturen betrachtet werden [77].

3.3 Ausgewählte Metadatenschemata

Metadatenschemata existieren für nahezu alle Anwendungsfälle, aufgrund der spezifischen Anforderungen einzelner Fachdisziplinen haben sich jedoch insbesondere generische Schemata als fachübergreifende Standards etabliert. In den folgenden Abschnitten werden zwei solcher Schemata kurz vorgestellt, gefolgt von einem speziell für die Ingenieurwissenschaften konzipierten Schema.

3.3.1 Dublin Core

Bei Dublin Core [83] handelt es sich um ein von der Dublin Core Metadata Initiative (DCMI) entwickeltes und interdisziplinäres Metadatenschema zur allgemeinen Beschreibung von Ressourcen im Internet. In seiner ursprünglichen Form, die wegen ihrer Einfachheit immer noch weit verbreitet ist, besteht das Schema aus lediglich 15 Kernelementen. Diese beinhalten sowohl typische, administrative Metadaten, wie z. B. Titel, Autor und Beschreibung, als auch technische Metadaten, wie den Typ oder das Format der beschriebenen Ressource. Die sogenannten DCMI Metadata Terms spezifizieren über diesen Kern hinaus zusätzliche Elemente für detailliertere Angaben, während das DCMI Type Vocabulary ein kontrolliertes Vokabular für unterschiedliche Ressourcentypen bereitstellt, das bei Verwendung von Dublin Core oder anderen Schemata eingesetzt werden kann. Die innerhalb von Dublin Core definierten Metadaten verwenden RDF als Beschreibungssprache und lassen sich somit in verschiedenen Formaten serialisieren, sind jedoch ebenfalls in anderen Kontexten nutzbar. Im Gegensatz zu anderen Schemata sind alle Elemente optional, können mehrfach vorkommen und in beliebiger Reihenfolge stehen.

3.3.2 DataCite Metadata Schema

Ein weiterer, ebenfalls sehr verbreiteter Metadatenstandard ist das DataCite Metadata Schema [84], das hauptsächlich vom DataCite-Konsortium im Rahmen des

gleichnamigen Dienstes entwickelt wird. Im Gegensatz zu Dublin Core ist das DataCite Metadata Schema auf einen bestimmten Anwendungsbereich ausgelegt: die bibliografische Beschreibung von Forschungsdaten, um diese mithilfe von PIDs zu kennzeichnen, wobei zunehmend auch fachliche Metadaten spezifiziert werden können. Als Identifikatoren kommen aktuell ausschließlich DOIs zum Einsatz, welche direkt über die von DataCite angebotenen Schnittstellen zusammen mit den entsprechenden Metadaten registriert werden können. Da es sich dennoch um ein interdisziplinäres Schema handelt, finden sich etliche Parallelen zu Dublin Core, wodurch ein Großteil der im DataCite Metadata Schema definierten Elemente mit den DCMI Metadata Terms interoperabel sind [85]. Die Elemente des DataCite Metadata Schema selbst sind in drei Kategorien gegliedert: obligatorische Metadaten, die immer bereitgestellt werden müssen, sowie empfohlene und optionale Metadaten. Die Metadaten werden klassischerweise in XML angegeben, das anhand eines bestehenden XML Schemas validiert wird, jedoch wird mittlerweile auch die Konvertierung von z. B. JSON-basierten Formaten unterstützt.

3.3.3 EngMeta

Während die bisher vorgestellten Metadatenschemata interdisziplinär sind, handelt es sich bei EngMeta [86] um ein fachspezifisches Beispiel. Der Fokus des Schemas liegt auf den computergestützten Ingenieurwissenschaften, es eignet sich jedoch auch für experimentelle Arbeitsabläufe. Ziel ist es, neben der Beschreibung der Forschungsdaten selbst, alle in einem Forschungsprozess beteiligten Komponenten festzuhalten, die zur Erstellung eines Datensatzes beigetragen haben, z. B. Instrumente oder Software, sowie deren Parameter bzw. Umgebungsbedingungen. Neben den fachlichen Metadaten werden jedoch auch hier allgemein Informationen wie Titel und Beschreibung der Datensätze erfasst. Um ein möglichst hohes Maß an Interoperabilität zu gewährleisten, basiert EngMeta auf unterschiedlichen, vorhandenen Schemata und Ontologien, darunter auch Dublin Core und das DataCite Metadata Schema. Wie bei letzterem wird das gesamte Metadatenmodell von EngMeta primär durch ein XML Schema spezifiziert.

3.4 Ausgewählte Ontologien

Ähnlich wie bei den vorgestellten Metadatenschemata haben sich vor allem fachübergreifende Ontologien etabliert. Ein solches Beispiel stellt auch die Ontologie bzw. das Datenmodell PROV [87] dar. Bei diesem handelt es sich um einen Standard des World Wide Web Consortiums (W3C) zur strukturierten Angabe der Herkunft von digitalen oder physischen Objekten, die als Entitäten bezeichnet werden. Entitäten werden von Aktivitäten erzeugt, die bestimmte Aktionen oder Prozesse beschreiben und dabei ebenfalls andere Entitäten verwenden oder transformieren können. Aktivitäten werden wiederum Agenten zugeordnet, welche für die jeweilige Aktivität, und damit die Erzeugung entsprechender Entitäten, verantwortlich sind. Hierbei muss es sich nicht unbedingt um Personen handelt, sondern z. B. auch um Software oder Gerätschaften. Neben dem konzeptuellen Datenmodell von PROV existiert eine in OWL-Syntax spezifizierte Ontologie sowie ein zum Datenmodell passendes XML Schema. PROV wird häufig zur Angabe der Herkunft von Forschungsdaten verwendet und für den interoperablen Austausch dieser Informationen in verschiedenen Umgebungen. Um aus bereits vorhandenen Metadaten die Herkunft der zugrunde liegenden Daten angeben zu können, sind unterschiedliche Konvertierungsmöglichkeiten spezifiziert, z. B. ausgehend von Dublin Core [88]. Auch in EngMeta [86] sind Möglichkeiten für eine entsprechende Konvertierung vorgesehen.

Eine sehr ähnliche Entwicklung ist das Open Provenance Model (OPM) [89]. Als Grundkomponenten definiert OPM Artefakte, auf die bestimmte Prozesse angewandt oder die von diesen erzeugt werden. Wie auch bei PROV sind die sogenannten Agenten für diese Prozesse verantwortlich oder zuständig. Das Datenmodell selbst ist ebenfalls abstrakt gehalten und beinhaltet unterschiedliche, konkrete Anwendungsformate, u. a. in Form einer OWL-Ontologie und eines XML Schemas.

3.5 Vokabulare

Neben Metadatenschemata und Ontologien sind ebenfalls Entwicklungen wie Schema.org [90], RDF Schema (RDFS) [91] oder das Data Catalog Vocabulary (DCAT) [92] erwähnenswert. Ähnlich wie beim bereits erläuterten DCMI Type Vocabulary handelt es sich bei diesen in erster Linie um kontrollierte Vokabulare, um z. B. die Semantik von Metadaten beschreiben zu können oder die Interoperabilität zwischen strukturierten Daten auf Websites und Datenkatalogen zu verbessern. Im Allgemeinen besteht eine große Schnittmenge zwischen Vokabularen, Metadatenschemata und Ontologien. So können z. B. einzelne Metadatenelemente aus Schemata wie Dublin Core als Vokabular genutzt werden, um in anderen Schemata oder Ontologien verwendet zu werden. Prinzipiell können Vokabulare ebenfalls innerhalb von Terminologieservices hinterlegt und referenziert werden.

3.6 Ausgewählte Forschungsdatenmanagementsoftware

In den folgenden Abschnitten werden konkrete Systeme und Softwarewerkzeuge vorgestellt, die sich jeweils den in Kapitel 2 erläuterten Arten von FDM-Software zuordnen lassen und primär im Forschungsumfeld eingesetzt werden. Bei sämtlichen Beispielen handelt es sich um quelloffene und größtenteils auch webbasierte Software, wodurch in den meisten Fällen eine zentrale Installation und Administration möglich ist.

3.6.1 Datenmanagementplan-Software

Wie bereits in Kapitel 2 erwähnt, können DMPs mithilfe entsprechender Software konzipiert werden. Ein in Deutschland populär gewordenes Werkzeug stellt der sogenannte Research Data Management Organiser (RDMO) [44] dar. RDMO

unterstützt die Erstellung von DMPs durch die Beantwortung von Fragen unterschiedlicher Kataloge, die sich z. B. an den konkreten Anforderungen von Förderorganisationen orientieren können, wobei auch die Definition eigener Fragenkataloge möglich ist. Die erstellten DMPs können anschließend in unterschiedlichen Formaten exportiert werden. Ein sehr ähnliches Werkzeug, das primär im US-amerikanischen Forschungsumfeld zum Einsatz kommt, stellt DMPTool [42] dar.

3.6.2 Elektronische Laborbücher

Ein insbesondere im europäischen Raum eingesetztes ELN stellt eLabFTW [93] dar. Der Fokus von eLabFTW liegt auf der Beschreibung von Experimenten durch die Angabe grundlegender und teilweise generischer Metadaten, Freitextbeschreibungen, Skizzen sowie einem Experiment zugehörige Dateianhänge. Experimente können in mehrere Versuchsschritte aufgeteilt werden und mit Proben, Gerätschaften oder anderen, digitalisierten Objekten aus einer zentralen Datenbank verknüpft werden. Während die Nutzung der beschriebenen Funktionalitäten hauptsächlich durch eine GUI erfolgt, ist ebenfalls eine webbasierte, auf HTTP aufbauende Programmierschnittstelle (englisch: Application Programming Interface, kurz API) vorhanden.

Ein weiteres Beispiel, das im Gegensatz zu eLabFTW stark fachspezifisch und auf den Einsatz in der Chemie ausgelegt ist, stellt Chemotion [94] dar. Durch den Fokus auf eine bestimmte Fachdisziplin ermöglicht Chemotion die Nutzung spezifischer Funktionalitäten, wie beispielsweise die Spezifizierung von Molekülen, die Teil von mehreren Proben sein können, oder von chemischen Reaktionen mithilfe eines entsprechenden Editors. Durch Anbindungen an chemische Datenbanken, wie z. B. PubChem [95], lassen sich die dadurch erfassten Informationen und Metadaten komplementieren. Die Besonderheiten von Chemotion liegen in der Anbindung von Laborgeräten über unterschiedliche Mechanismen [51], sowie in der Integration eines zentralen Repositoriums [96]. Letzteres wurde speziell

im Kontext von Chemotion entwickelt und ist dementsprechend ebenfalls auf die Chemie spezialisiert.

3.6.3 Repositorien

Eines der bekanntesten Repositorien stellt Zenodo [97] dar, das von der Europäischen Organisation für Kernforschung (CERN) entwickelt und zentral betrieben wird. Zenodo dient der öffentlichen Publizierung beliebiger Forschungsdaten und kommt insbesondere auch vermehrt bei Software zum Einsatz, mithilfe einer entsprechenden Integration mit GitHub [98]. Zur Vergabe von DOIs kommt DataCite zum Einsatz, weshalb das von Zenodo verwendete Metadatenschema sich stark am DataCite Metadata Schema orientiert.

Ein weiteres Beispiel ist Dataverse [99]. Konkrete Installationen von Dataverse sind weltweit im Einsatz, die bekannteste Instanz ist das Harvard Dataverse [100]. Bei dem Repositorium handelt es sich um eine generische Entwicklung zur Publizierung von Forschungsdaten und Metadaten in sogenannten Dataverses. Jedem Dataverse können eigene Informationen, Benutzerrechte und Metadatenschemata zugeordnet werden, wobei letztere aus einer vordefinierten Auswahl generischer oder fachspezifischer Schemata ausgewählt werden können. Auch die Definition benutzerdefinierter Schemata ist möglich, was z. B. für EngMeta innerhalb eines auf Dataverse aufbauenden, institutionellen Repositoriums der Universität Stuttgart umgesetzt wurde [86].

Zur Suche von passenden Daten für die Planung eines Forschungsvorhabens sind unterschiedliche Plattformen im Einsatz, wie z. B. re3data [101] oder DFG RIsources [102]. Hierbei handelt es sich um spezielle Arten von Meta-Repositorien, die zur Suche von anderen, öffentlich zugänglichen Repositorien verwendet werden können. Neben der Suche nach fachspezifischen Systemen, wie z. B. dem Repositorium NOMAD [103] für die computergestützten Materialwissenschaften, können Aspekte wie Metadatenschemata, Möglichkeiten zur Angabe von Lizenzen oder Zertifizierungen berücksichtigt werden.

3.6.4 Workflow-Management-Systeme

Workflow-Management-Systeme sind in verschiedensten Ausprägungen vorhanden und unterscheiden sich stark in deren Ausrichtung, Benutzerfreundlichkeit und Flexibilität. Zwei nennenswerte Beispiele stellen Galaxy [104] und AiiDA [105] dar. Bei beiden Systemen handelt es sich um größtenteils generische Entwicklungen, wobei Galaxy insbesondere auf die Lebenswissenschaften spezialisiert ist. Während Galaxy eine webbasierte Plattform zur Verfügung stellt, um mithilfe einer GUI unterschiedliche Werkzeuge einzeln oder kombiniert als Workflow ausführen zu können, sind zur Verwendung von AiiDA Programmierkenntnisse und vorherige Konfigurationen zur Ausführung von Workflows notwendig. Der Fokus liegt dafür stärker auf der Angabe der Datenherkunft, da sämtliche Zwischenschritte und Parameter als Graph in einer dafür vorgesehenen Datenbank hinterlegt werden.

3.6.5 Registries und Terminologieservices

Wie bereits in Kapitel 2 erwähnt, sind im Kontext von Registries insbesondere Terminologieservices zur Ablage und Referenzierung von Konzepten und Termen für diese Arbeit relevant. Ein entsprechender Dienst wird im deutschen Forschungsumfeld z. B. von der Technischen Informationsbibliothek (TIB) angeboten [106]. Dieser stellt primär eine Suchfunktion bereit, die sowohl über eine GUI als auch über eine webbasierte HTTP-API genutzt werden kann. Zur Gruppierung fachspezifischer Terminologien werden zudem unterschiedliche Rubriken angeboten, wobei u. a. eine auf die Ingenieurwissenschaften ausgelegte Rubrik im Rahmen von NFDI4Ing bereitgestellt wird.

Eine weitere, nennenswerte Art von Registry stellt die Open Researcher and Contributor ID (ORCID) [107] dar. Bei dieser handelt es sich um einen PID zur eindeutigen Kennung von Forschern, wobei Informationen über assoziierte Institutionen oder Publikationen mithilfe eines dazugehörigen Dienstes zentral hinterlegt und abgerufen werden können.

4 Konzepte

Anhand der Vielzahl an unterschiedlichen, in Kapitel 3 vorgestellten Initiativen, Konzepten und Technologien wird die Schwierigkeit des FDMs in den Ingenieurwissenschaften deutlich, die durch die Heterogenität der verschiedenen Fachdisziplinen und entsprechender Forschungsprozesse begründet ist. Um einen Beitrag zur Verbesserung des Status quo zu leisten, lässt sich prinzipiell an mehreren Punkten anknüpfen, etwa bei der Etablierung von Datenkompetenzen, der Konzeption von Metadatenstandards und Ontologien, oder auch bei der (Weiter-)Entwicklung von FDM-Software. In dieser Arbeit liegt der Schwerpunkt auf dem letztgenannten Ansatz. Während Aspekte wie Datenkompetenzen bereits während der Ausbildung zukünftiger Forscher eine wichtige Rolle spielen können, ist ein strukturiertes FDM ohne eine technische Infrastruktur und entsprechender Software nicht umzusetzen [60, 66]. Gleiches gilt für die konkrete und vor allem praxisorientierte Nutzung von Metadatenstandards oder Ontologien. In diesem Kapitel werden daher die Ziele, entsprechende Konzepte und mögliche Implementierungsansätze eines Systems zur Unterstützung des FDMs mit Fokus auf den Ingenieurwissenschaften evaluiert und definiert. Diese wird dabei konzeptuell als ein Gesamtsystem betrachtet, jedoch kann die konkrete Implementierung prinzipiell aus mehreren, unabhängigen Systemen bestehen und insbesondere auch die Weiterentwicklung oder Verwendung existierender Software unter Verwendung geeigneter Schnittstellen umfassen.

4.1 Zielsetzung des Systems

Unabhängig von der angestrebten Fachdisziplin sollten die FAIR-Prinzipien bei der Konzeption eines FDM-Systems berücksichtigt werden, um sicherzustellen, dass mithilfe des Systems verwaltete Forschungsdaten und Metadaten auffindbar, maschinenlesbar und wiederverwendbar sind. Eine der Grundvoraussetzungen der Prinzipien und entsprechender Architekturen ist die Zugänglichkeit und Referenzierbarkeit von Datensätzen durch PIDs [34], wobei Datensätze prinzipiell als eigenständige Entitäten digitaler oder digitalisierter Objekte betrachtet werden können [60]. Metadaten müssen als zusätzliche Komponente gleichermaßen spezifizierbar sein, um durch passende Verknüpfungen mit den Datensätzen deren FAIRness zu erhöhen sowie die Herkunft der Daten nachvollziehbar machen zu können. Dies kann ebenfalls die Auffindbarkeit über entsprechende Suchmaschinen und die Wiederverwendbarkeit des gesamten Datensatzes ermöglichen, idealerweise unter Angabe entsprechender Lizenzen nur Nachnutzung. Die Interoperabilität des Datensatzes kann dabei, sofern möglich, durch die Verwendung von Metadatenstandards und offenen Dateiformaten verbessert werden. Das konzipierte System sollte daher eine solche, schrittweise FAIRness von Datensätzen ermöglichen, jedoch gleichzeitig nicht die Notwendigkeit zur „vollständigen“ FAIRness eines jedes Datensatzes erzwingen, da eine teilweise FAIRness je nach konkretem Anwendungsfall ausreichen kann oder z. B. aufgrund fehlender Metadatenschemata nicht immer möglich ist. Insbesondere bedeutet FAIR nicht zwangsläufig, dass entsprechende Datensätze auch frei und öffentlich zugänglich sind [34], z. B. im Sinne der Vorstellung von Open Data, weshalb ebenfalls Zugriffsbeschränkungen für individuelle Datensätze relevant sein können.

Da die FAIR-Prinzipien absichtlich keine technischen Anforderungen spezifizieren, sondern lediglich eine Reihe von Leitlinien darstellen, die über verschiedene Implementierungen und Systeme hinweg Bestand haben können, sollten diese jedoch nicht als Standard gesehen werden [34]. Weiterhin liegt der Fokus der Prinzipien konzeptuell auf den letzten Phasen des Forschungsdatenlebenszyklus: der Publizierung, Archivierung und Nachnutzung von Forschungsdaten. Gleichzeitig sollten daher ebenfalls die verbleibenden Phasen im konzipierten

System berücksichtigt werden und sich dabei an den praktischen Anforderungen unterschiedlicher Forscher und deren Bedürfnissen an das Datenmanagement orientieren. Bereits bei der Planung eines Forschungsvorhabens kann FDM-Software etwa durch die Konzeption oder Integration von DMPs unterstützen. Die wichtigsten Phasen des Forschungsdatenlebenszyklus stellen jedoch die Erhebung und Aufbereitung von Daten und Metadaten selbst dar, da hier die Durchführung des eigentlichen Forschungsprozesses stattfindet. In den Ingenieurwissenschaften kommen sowohl experimentelle, simulative als auch hybride Anwendungsfälle zum Einsatz. Computer können dabei als primärer Datenerzeuger oder lediglich als Hilfsmittel, z. B. zur Steuerung eines Geräts, eingesetzt werden. Für das konzipierte System setzen solch heterogene Datenquellen und damit -formate eine stark generische Entwicklung voraus, was ebenfalls die Etablierung von Schnittstellen zur Verwendung und Automatisierung bestehender Software oder Laborgeräte umfassen kann. Dieser Aspekt ist ebenfalls für die mögliche Integration existierender FDM-Software relevant, wobei generell der Fokus auf der Komplementierung bereits etablierter Arbeitsabläufe unter potenzieller Nutzung solcher Software liegen sollte.

Anhand dieser Anforderungen ist für das System eine webbasierte Implementierung zu bevorzugen, welche die einfache, Webbrowser-basierte Nutzung ohne die Installation zusätzlicher Software sowie eine zentrale Administration auf unterschiedlichen Ebenen ermöglicht, z. B. innerhalb einzelner Arbeitsgruppen oder gesamter Institutionen. Das System orientiert sich dabei an einer Vielzahl bereits existierender, webbasierter FDM-Systeme. Neben einer benutzerfreundlichen GUI sollte ebenfalls eine entsprechende, webbasierte HTTP-API bereitgestellt werden, um das bereits erwähnte Potenzial zur Automatisierung und zur Integration existierender Software zu ermöglichen. Prinzipiell kann hierbei lediglich die Nutzung innerhalb nicht an ein zentrales Netzwerk angeschlossener Laborrechner oder -geräte eine Schwierigkeit darstellen [51], was jedoch ein generelles Problem webbasierter Systeme ist. Die einfache Nutzung und Administration des konzipierten Systems ist auch deswegen wichtig, da auf lange Sicht tendenziell die Verantwortung des strukturierten FDMs weiterhin bei individuellen Institutionen [70] liegen wird, wie bereits in Kapitel 3 erwähnt. Eine, zumindest von nationaler oder

internationaler Ebene aus betrachtet, dezentralisierte Forschungsdateninfrastruktur mit gemeinsamen, interoperablen Datenmodellen, Standards und Schnittstellen stellt daher eine realistische Zukunftsperspektive dar [67, 108], in die sich das konzipierte System eingliedern können muss.

4.2 Strukturierte Verwaltung von Metadaten

Metadaten stellen als spezielle Form der Datendokumentation einer der wichtigsten Aspekte eines strukturierten FDMs zur Auffindbarkeit und Wiederverwendbarkeit von Forschungsdaten dar. Da Metadaten in sämtlichen Arten von datengestützten Forschungsprozessen relevant sein können, wird die Verwaltung von diesen in den folgenden Abschnitten als separater Aspekt des konzipierten Systems betrachtet. Die hierbei evaluierten Konzepte sind größtenteils generisch und lassen sich dementsprechend ebenfalls auf die heterogenen Metadatenbedarfe der Ingenieurwissenschaften anwenden.

4.2.1 Metadatenstandards und -schemata

Eine zentrale Frage, die sich bei der Konzeption einer konsistenten Metadatenstruktur stellt, ist, ob ein oder mehrere existierende Metadatenstandards bzw. -schemata eingesetzt werden können, wie auch innerhalb der FAIR-Prinzipien gefordert. Kennedy [109] schlägt neun praxisorientierte Fragen vor, die sich Entwickler von Sammlungen digitaler Objekte stellen können, um bei dieser Fragestellung zu unterstützen, betont aber gleichzeitig, dass kein Schema existiert, dass alle Vorhaben und Domänen auf einmal abdeckt. Die wichtigste Frage betrifft die externe Nutzung der Metadaten, was deren Auffindbarkeit und Potenzial zur Wiederverwendbarkeit für die angestrebte Zielgruppe umfasst. Je stärker das gewählte Metadatenschema den Erwartungen der Nutzer entspricht, desto einfacher lassen sich diese Ziele erreichen, unabhängig von der tatsächlichen, technischen Umsetzung. Der eigentliche Prozess der Spezifikation von Metadaten kann ebenfalls die Wahl des Schemas beeinflussen, insbesondere in Bezug auf die

daran beteiligten Personen. Diese werden von Kennedy als separate Katalogisierer betrachtet, jedoch kann es sich in der Praxis auch um dieselbe Personengruppe handeln, welche die Metadaten später nutzen möchte. Zuletzt spielt die Interoperabilität eine wichtige Rolle, wobei die Relation zu anderen digitalen Sammlungen und entsprechender Metadatenschemata, aber auch die Flexibilität des Schemas berücksichtigt wird.

Duval et al. [110] dagegen betrachten Schemata weniger aus der Perspektive der Nutzer, sondern definieren verschiedene Prinzipien für deren Konzeption, die sich allgemein auf Metadaten unterschiedlicher Disziplinen anwenden lassen können. In Bezug auf Metadatenschemata spielen insbesondere die Modularität und Erweiterbarkeit eine Rolle. Modularität spricht dabei nicht zwangsläufig gegen die Verwendung eines fixen Metadatenschemas, sondern umfasst auch die Konzeption neuer Schemata unter Verwendung bestehender Standards. Gleichzeitig kann die Verwendung fixer Schemata die Erweiterbarkeit bzw. Flexibilität einer Metadatenstruktur beeinflussen. Diese ist dann notwendig, wenn z. B. neue Metadatenelemente hinzugefügt oder existierende Elemente verfeinert werden sollen, um damit domänenspezifische Bedarfe abzudecken. Um ein gewisses Maß an Interoperabilität in allen Fällen zu gewährleisten, kann ein Basisschema einen Grundbestand an gemeinsamen Metadaten spezifizieren. In der Praxis lassen sich diese Anforderungen z. B. mithilfe sogenannter Application Profiles (APs) [111] realisieren. Bei diesen handelt es sich um pragmatische Kombinationen existierender Metadatenschemata, die für eine bestimmte, lokale Anwendung optimiert sind. Während vorhandene Elemente verfeinert oder deren Werte eingeschränkt werden können, dürfen jedoch laut Definition keine neuen, nicht aus vorhandenen Schemata stammenden Elemente hinzugefügt werden. Ein Beispiel eines Application Profiles, das innerhalb des European Data Portals (EDP) [112] zur einheitlichen Beschreibung von hauptsächlich administrativen Metadaten zum Einsatz kommt, ist DCAT-AP [113]. Dieses baut größtenteils auf dem durch DCAT bereitgestellten Vokabular auf und erweitert dieses um zusätzliche Terminologie.

4.2.2 Metadatenqualität

Während in der Vergangenheit viele Arbeiten auf die Spezifikation neuer Metadatenschemata fokussiert waren, die bereits deren Struktur, Syntax und teilweise auch Semantik definieren können, bleibt dennoch die Frage bestehen, was gute Metadaten überhaupt ausmacht und wie sich deren Qualität messen lässt. Die FAIR-Prinzipien beinhalten die Beschreibung von Daten mit „umfangreichen“ Metadaten [8], gehen jedoch nicht ins Detail, welche Anforderungen solche Metadaten erfüllen müssen. Lange vor der Veröffentlichung der FAIR-Prinzipien wurden bereits Metriken zur Messung der Qualität von Metadaten definiert [114–116]. Park [117] fasst einige dieser Metriken unter den darin am häufigsten angewandten Begriffen Vollständigkeit, Genauigkeit und Konsistenz zusammen:

- Vollständigkeit: Die Vollständigkeit von Metadaten gibt an, wie detailliert ein Objekt unter Berücksichtigung der Praktikabilität beschrieben wird. Der Grad der Vollständigkeit richtet sich jeweils nach dem Typ des beschriebenen Objekts und den bewährten Praktiken der jeweiligen Domäne, ist also vom Kontext abhängig. Bei Verwendung eines Metadatenschemas bedeutet Vollständigkeit nicht zwangsläufig, dass sämtliche Elemente des Schemas verwendet werden müssen.
- Genauigkeit: Genauigkeit bedeutet nicht nur, dass der Inhalt von Metadaten akkurat, korrekt und sachlich ist, sondern auch frei von z. B. Tippfehlern oder nicht etablierten Abkürzungen.
- Konsistenz: Die Konsistenz von Metadaten kann sich sowohl auf deren Syntax als auch Semantik beziehen. Auf syntaktischer Ebene lassen sich Inkonsistenzen oft auf verschiedene Schreibweisen gleichartiger Metadatenelemente zurückführen, z. B. unterschiedliche Datumsformate. Die semantische Konsistenz bezieht sich dagegen darauf, dass gleiche Elemente für dieselben oder zumindest ähnliche Konzepte verwendet werden, z. B. in Form von standardisierten Identifikatoren, die für verschiedene Arten von Seriennummern oder Bezeichnern eingesetzt werden können.

Die Genauigkeit und Konsistenz spielen vor allem bei der Interoperabilität von Metadaten eine wichtige Rolle. Entsprechende Schwierigkeiten zeigen sich bei der Zuordnung von Metadaten aggregierter Datensammlungen, selbst bei Verwendung einheitlicher Metadatenschemata [118], und sind im Falle heterogener und multidisziplinärer Datenquellen noch verschärft. Zwar korreliert in Bezug auf die reine Anzahl von Zugriffen und Downloads die Qualität der Metadaten nicht unbedingt mit der Nachnutzung von dadurch beschriebenen und publizierten Forschungsdaten [119, 120], dennoch ist die Metadatenqualität wichtig, um die Nachnutzung der Forschungsdaten überhaupt erst ermöglichen zu können. Diese Ansicht wird auch von Donaldson und Koepke [121] geteilt, die Forscher verschiedener Disziplinen zu deren Wünschen bei der Publizierung von Forschungsdaten in Repositorien befragt haben, wobei Qualitätskontrolle eine zentrale Rolle spielt.

Park [117] schlägt zur Verbesserung der Metadatenqualität zwei grundlegende Mechanismen vor: die Definition und Nutzung von Richtlinien und die automatische Generierung von Metadaten mithilfe entsprechender Software. Richtlinien können, aufbauend auf Metadatenschemata, zusätzliche Hilfestellungen zur Angabe von Metadaten bieten, z. B. textuelle Beschreibungen, Beispiele oder Regeln für die Kodierung einzelner Metadatenelemente. Hierbei bietet es sich an, entsprechende Richtlinien innerhalb vorhandener Systeme wie z. B. Repositorien einzubetten, unter Verwendung simpler Formularelemente wie Dropdowns zur Auswahl vorgegebener Werte oder Pop-ups mit zusätzlichen Informationen [122]. Bei der automatischen Generierung von Metadaten lässt sich zwischen zwei grundlegenden Arten von Software unterscheiden: primär zur Generierung von Forschungsdaten verwendete Software und spezialisierte Software zur Generierung von Metadaten [123]. Erstere Art bietet sich häufig zumindest zur Erfassung technischer Metadaten an, wie z. B. Erstellungsdaten oder Dateigrößen, muss jedoch nicht darauf beschränkt sein. Je nach Software können Metadaten separat vorliegen oder in die erzeugten Daten eingebettet sein, wobei es in beiden Fällen wichtig ist, dass Metadaten in standardisierte und strukturierte Datenformate überführt werden können. Mithilfe von spezialisierter Software kann dagegen die Generierung von Metadaten anhand existierender, digitaler Objekte erfolgen, wobei nicht nur der Inhalt der Objekte, sondern auch deren Beziehungen und

Kontext einbezogen werden können [124, 125]. Zwar ist der Nutzen solcher Software limitiert, insbesondere im Hinblick auf die Metadatenqualität [126], jedoch können (semi-)automatisch generierte Metadaten in einem kombinierten Ansatz ebenfalls als Vorschläge bei der manuellen Angabe von Metadaten zum Einsatz kommen [124].

4.2.3 Ontologien und Datenherkunft

Zur Integration von in Ontologien spezifizierten Konzepten bieten sich Terminologieservices an, die zur Interoperabilität von Metadatenelementen und damit ebenfalls zur Verbesserung der Metadatenqualität beitragen können [110, 127]. Insbesondere die semantische Abbildung von Metadatenelementen zweier Schemata wird dadurch ermöglicht, was häufig auch als sogenannter Crosswalk bezeichnet wird [128]. Eine Schwierigkeit, die sich bei Verwendung von Terminologieservices prinzipiell stellen kann, ist die korrekte Auswahl passender Konzepte für die jeweiligen Metadatenelemente. Dies ist dadurch begründet, dass die bereitgestellten Terme oft aus einer Vielzahl unterschiedlicher Ontologien bzw. Domänen und Einsatzgebiete stammen, weshalb gleichlautende Terme unterschiedliche Konzepte beschreiben können [61]. Selbst bei der Verwendung fixer Terme innerhalb von Metadatenschemata kann dies spätestens bei der Aggregation mehrerer Datenquellen zu Problemen führen, insbesondere bei der Verwendung multidisziplinärer Schemata.

Da ein zentrales Merkmal von Ontologien die Angabe von Relationen darstellt, ist deren Verwendung auch bei der Spezifikation der Datenherkunft relevant. Diese spielt eine wichtige Rolle für die Wiederverwendbarkeit von Forschungsdaten als Teil der FAIR-Prinzipien, da erst mithilfe der Datenherkunft der gesamte Forschungsprozess, der Daten und Metadaten zugrunde liegen kann, reproduzierbar wird. Im Kontext des FDMs spielen hierbei weniger die konkreten Metadatenelemente der Forschungsdaten eine Rolle, sondern die Beziehungen mehrerer Datensätze untereinander. Zur gezielten Spezifikation der Datenherkunft wurden zwei konkrete Ontologien bzw. Datenmodelle in Kapitel 3 bereits vorgestellt.

Ein im Gegensatz zu diesen stark praxisorientierter Ansatz stellt das sogenannte W7-Modell [129] dar, dem die sieben Fragen *Was* (Ereignis), *Wann* (Zeitpunkt), *Wo* (Ort), *Wie* (Aktion), *Wer* (Akteur), *Welches* (z. B. Software oder Gerät) und *Warum* (Grund) zugrunde liegen, in Bezug auf unterschiedliche Ereignisse, die z. B. zur Erstellung oder Modifikation eines Datensatzes beitragen. Das Modell lässt sich vergleichsweise einfach auf bereits vorhandene Forschungsdaten und Metadaten durch Beantwortung der sieben Fragen anwenden, was u. a. bereits für die (semi-)automatische Extraktion von Informationen zur Datenherkunft aus existierenden Einträgen innerhalb eines ELNs erfolgt ist [130].

4.2.4 Metadatenformate

Neben der Struktur und dem Inhalt von Metadaten spielt zuletzt auch das Format eine Rolle, das zur Persistierung der Metadaten verwendet wird. Da das primäre Ziel von Metadaten die Beschreibung digitaler Objekte ist, und es sich hierbei oft um Forschungsdaten im klassischen Sinne handelt, besteht ein möglicher Ansatz in der Einbettung von Metadaten in den Forschungsdaten selbst. In Kapitel 2 wurde bereits der Standard XMP erwähnt, der insbesondere bei Bilddaten häufig zum Einsatz kommt [131], durch seine Beschränkung auf bestimmte Datenformate jedoch nicht für den Einsatz heterogener Forschungsdaten geeignet ist. Eine mögliche Alternative stellt das Datenformat HDF5 dar, das bereits nativ die Spezifizierung von Metadaten zulässt, jedoch in dieser Hinsicht auf einfache Schlüssel/Wert-Paare beschränkt ist.

Zur separaten Speicherung von Metadaten bieten sich strukturierte Formate an, wobei aufgrund ihrer Verbreitung insbesondere JSON oder XML als syntaktische Grundlage in Frage kommen. Aufgrund seiner Einfachheit, des geringen Speicherbedarfs und der leichten Lesbarkeit für Menschen und Maschinen, ist JSON mittlerweile zum Quasistandard für den Austausch von Daten im Web geworden [132]. Gleichzeitig bietet XML zusätzliche Funktionalitäten wie Namensräume, Attribute und eine standardisierte Unterstützung von Schemata, ist insgesamt also eigenständiger. Neben den technischen Aspekten spielt jedoch

auch die Verwendung bestehender Metadatenschemata wie z. B. dem DataCite Metadata Schema oder EngMeta eine Rolle. Beiden Schemata liegt jeweils ein XML Schema als Basis zugrunde, weshalb die Angabe entsprechender Metadaten in XML zu bevorzugen ist, jedoch wird die Verwendung von JSON und entsprechender Schemata innerhalb von DataCite zunehmend ausgebaut. In Dublin Core hingegen wird RDF als Beschreibungssprache eingesetzt, wodurch ebenfalls verschiedene Serialisierungsformate möglich sind. Auch wenn keines der genannten Schemata im konzipierten System direkt eingesetzt werden sollte, muss zumindest die Interoperabilität auf syntaktischer Ebene berücksichtigt werden, weshalb eine Orientierung an den jeweils verwendeten Formaten sinnvoll ist.

4.2.5 Zusammenfassung

Bei der Konzeption einer geeigneten Struktur, welche die Nutzer bei der Spezifikation und Verwaltung von Metadaten unterstützt, sind zusammenfassend etliche Aspekte zu beachten, angefangen mit der möglichen Verwendung eines oder mehrerer geeigneter Metadatenschemata. Diese sollten auf die angestrebte Zielgruppe abgestimmt sein und gleichzeitig ein größtmögliches Maß an Interoperabilität gewährleisten. Da es sich bei den Ingenieurwissenschaften um eine vergleichsweise heterogene Landschaft an unterschiedlichen Forschungsdisziplinen handelt, liegt die Vermutung nahe, dass kein einzelnes Schema die Erwartungen aller potenziellen Nutzer abdecken kann. Ein möglicher Ansatz stellt das Schema EngMeta dar. Dieses zeichnet sich durch seine Modularität aus, welche durch die Verwendung existierender Schemata und Ontologien ermöglicht wird, was ebenfalls eine partielle Interoperabilität mit diesen erlaubt. Damit lässt sich EngMeta im Grunde auch als Application Profile unterschiedlicher Schemata für die Ingenieurwissenschaften bezeichnen, bietet jedoch als fixes Schema nach wie vor wenig Flexibilität zur Abdeckung anwendungsspezifischer Bedarfe. Um diesen Aspekt zu berücksichtigen, stützt sich das in dieser Arbeit definierte Konzept stattdessen auf ein Basisschema interdisziplinärer Metadatenelemente, das als erweiterbare Grundlage für generische und domänenspezifische Metadaten dient. Dieses kann auf existierenden Schemata bzw. kontrollierten Vokabularen wie

Dublin Core oder Schema.org aufbauen und allgemeine Kernelemente wie z. B. den Titel und die Beschreibung eines digitalen Objekts oder technische Metadaten konkreter Forschungsdaten beinhalten. Zur Spezifikation generischer Metadaten können dagegen Templates zum Einsatz kommen, die eine flexible und modulare Version eines Metadatenschemas darstellen und als benutzerdefinierte Application Profiles betrachtet werden können. Zur Wiederverwendung existierender Metadatenelemente und entsprechender Terme können weiterhin Terminologieservices eingesetzt werden bzw. unabhängig davon die Spezifikation standardisierter Terme wie z. B. aus Schemata wie EngMeta ermöglicht werden. Da die Existenz geeigneter Terminologien jedoch nicht zwangsläufig gewährleistet werden kann, wird die Definition der Application Profiles für das konzipierte System um zusätzliche Flexibilität erweitert, um ebenfalls eigens definierte bzw. nicht standardisierte Metadatenelemente verwenden zu können.

Das Metadatenkonzept ähnelt damit den ursprünglichen Entwicklungen des Repositoriums Dryad [133]. Bei diesem wurde von Anfang an ein zweigleisiger Ansatz verfolgt, dessen Ziel es ist, sowohl kurzfristige Metadatenbedarfe abzudecken, als auch langfristig einsatzfähig und interoperabel zu bleiben [134]. Für die unmittelbaren Bedarfe kommt dabei ein stark an Dublin Core angelehntes Application Profile zum Einsatz, welches einfach in der Verwendung ist und ein hohes Maß an Interoperabilität gewährleistet. Die längerfristigen Bedarfe zielen dagegen insbesondere auf das Semantic Web ab, wofür die Metadaten in entsprechende RDF-Graphen überführt werden können. Ähnlich verhält sich das für diese Arbeit konzipierte Basisschema, das durch die Verwendung existierender Standards bereits ein gewisses Maß an Interoperabilität sicherstellt, aufgrund der Erweiterung um weitere Metadatenelemente jedoch ebenfalls spezifische Bedarfe von Forschern abdecken kann. Das Konzept ist somit ebenfalls stark an den praktischen und generischen Anforderungen der Ingenieurwissenschaften orientiert. Die Interoperabilität der Metadaten kann dabei, unter möglicher Verwendung standardisierter Terme, unterschiedlich stark ausgeprägt sein, was jedoch ebenfalls von der Existenz entsprechender Schemata oder Ontologien abhängt. In diesem Zusammenhang bietet ein generisches Metadatenkonzept langfristig ebenfalls

Möglichkeiten zur kollaborativen Bottom-up-Entwicklung neuer Metadatenschemata.

Da die Verwendung existierender Metadatenschemata zur Genauigkeit und Konsistenz von Metadaten beitragen kann, sollte bei einem solch flexiblen Metadatenkonzept besonders darauf geachtet werden, dass die Qualität der Metadaten weiterhin möglichst gewährleistet ist. Um zusätzliche Richtlinien für einzelne Metadatenelemente festlegen zu können, wird daher das Konzept der Templates um zusätzliche Validierungsoptionen erweitert, wie z. B. die Definition vorgegebener Werte, aus denen Anwender eines Templates wählen können. Solche Optionen lassen sich in webbasierten Applikationen ideal durch die Verwendung entsprechender Formularelemente einbinden. Weiterhin sollte, wo möglich, die automatische Generierung bzw. Extraktion von Metadaten möglichst nah an deren Quelle im Vordergrund stehen, um Fehler bei der manuellen Spezifikation von Anfang an gering halten zu können. Der Schwerpunkt liegt dabei auf der Erfassung solcher Metadaten, die bereits von Software oder Geräten als Teil bestehender Arbeitsabläufe erzeugt werden. In diesem Kontext spielt ebenfalls die Datenherkunft eine Rolle, die insbesondere bei der automatisierten Akquise von Daten und Metadaten realisiert werden kann, um verwandte Forschungsprozesse in Beziehung zu setzen. Das vorgestellte W7-Modell zeigt, dass sich viele der bei der Datenherkunft relevanten Fragen bereits oft anhand existierender Metadaten beantworten lassen, weshalb die Spezifikation der Datenherkunft in das bereits etablierte Metadatenkonzept eingegliedert werden sollte. Dazu müssen Möglichkeiten geschaffen werden, um digitale Objekte in Form geeigneter Metadatenelemente untereinander referenzieren zu können, wobei dies in Bezug auf das konzipierte System ebenfalls extern definierte Objekte umfassen kann. Um interoperabel mit Standards wie z. B. PROV zu bleiben, können Crosswalks zum Einsatz kommen [86], abhängig von den konkreten Metadatenelementen, welche den entsprechenden digitalen Objekten zugeordnet sind.

Unabhängig von der konzeptuellen Strukturierung der Metadaten bleibt zuletzt die Frage bestehen, in welcher Form diese persistiert und verwaltet werden sollen. Während unterschiedliche Technologien existieren, um Metadaten gemeinsam mit

den eigentlichen Forschungsdaten zu speichern, ist deren Nutzung entweder auf bestimmte Datenformate beschränkt, z. B. im Fall von XMP, oder auf bestimmte Metadatenstrukturen, z. B. bei der Verwendung von HDF5, das sich zudem durch eine vergleichsweise hohe Komplexität auszeichnet. Die separate Speicherung von Metadaten dagegen ermöglicht nicht nur die Nutzung gängiger Serialisierungsformate wie JSON oder XML, sondern erleichtert weiterhin die Zugänglichkeit im Hinblick auf die FAIR-Prinzipien, etwa durch die Bereitstellung der Metadaten mithilfe separater Datenbanken [135]. In Bezug auf konkrete Datenformate stellen etliche Schemata bestimmte Anforderungen an die Spezifikation von Metadaten, z. B. durch die Verwendung von XML Schemata. Das in dieser Arbeit etablierte Konzept sollte jedoch möglichst formatagnostisch sein, um eine syntaktische Interoperabilität mit unterschiedlichen Schemata und entsprechende Crosswalks ermöglichen zu können [128]. Eine zentrale Rolle als Zwischensprache kann hierbei RDF einnehmen, als formatagnostisches Modell zur Repräsentation von Metadaten und Beziehungen. RDF stellt eine natürliche Vorstufe zu Ontologien dar und dient als möglicher Anknüpfungspunkt zum Semantic Web oder der LDP-Architektur. Durch den einfachen Aufbau von RDF-Graphen in Form von Tripeln lassen sich darüber hinaus Metadaten verschiedener Schemata üblicherweise in RDF überführen [136]. Zuletzt sollte sich die Wahl eines innerhalb des konzipierten Systems verwendeten Metadatenformats an den praktischen bzw. technischen Rahmenbedingungen bezüglich Aspekten wie der Persistenz, systeminternen Anforderungen sowie den Konvertierungsmöglichkeiten in standardisierte Formate und Strukturen orientieren.

4.3 Strukturierte Verwaltung von Daten

Neben den Metadaten spielt auch die strukturierte Verwaltung der eigentlichen, potenziell mit den Metadaten verknüpften Forschungsdaten eine mindestens ebenso wichtige Rolle. Deren Verwaltung ist ebenfalls für sämtliche Arten datengestützter Forschungsprozesse relevant und wird daher in den folgenden Abschnitten, unter Berücksichtigung des bereits etablierten Metadatenkonzepts, separat aufgeführt.

Der Fokus liegt dementsprechend nicht ausschließlich auf der Publizierung oder Archivierung von Forschungsdaten, sondern auf den allgemeinen Anforderungen an die Datenverwaltung.

Bei der Entscheidung, wie Forschungsdaten gespeichert, verwaltet und bereitgestellt werden sollen, sind neben der Datenqualität ebenfalls verschiedene technische Aspekte zu berücksichtigen, wie das Datenvolumen, Methoden zur Datenübertragung und Dateisysteme zur Persistierung der Daten. Diese wiederum hängen von den konkreten Anforderungen der unterschiedlichen Forschungsdisziplinen und Arbeitsweisen in den Ingenieurwissenschaften ab, z. B. fallen bei der Durchführung von Simulationen in den Materialwissenschaften typischerweise größere Datenvolumen an, insbesondere im Bereich des High Performance Computings (HPC) [137]. Unabhängig davon werden Forschungsdaten häufig innerhalb unterschiedlicher Speichermedien, in heterogenen Formaten und an unterschiedlichen Orten gespeichert, was ebenfalls eine generische Umsetzung der Datenverwaltung erfordert.

4.3.1 Datenqualität und -formate

Die Qualität von Forschungsdaten selbst spielt insbesondere im Kontext von Repositorien eine Rolle, um die Nachnutzung der Daten ermöglichen zu können [138], allgemein gesehen jedoch auch innerhalb anderer Arten von FDM-Software, bei der die Verwaltung und vor allem (öffentliche) Bereitstellung von Daten möglich ist. Wie auch bei den Metadaten lassen sich unterschiedliche Qualitätsmerkmale für die eigentlichen Forschungsdaten definieren, wobei Dokumentationen und Metadaten zur Beschreibung der Daten das wichtigste Kriterium darstellen [139]. Metadaten stellen ihre beschreibenden Daten in den für Forscher notwendigen Kontext und erlauben daher eine individuelle Qualitätsbeurteilung, sofern wiederum die Qualität der entsprechenden Metadaten gegeben ist. Die Qualität von Forschungsdaten und Metadaten ist daher stark voneinander abhängig [138], weshalb dieser Aspekt ebenfalls im Zentrum der FAIR-Prinzipien steht [8].

Die eigentlichen Forschungsdaten betreffend ist vor allem deren Format für die Qualität relevant, auf das sich ebenfalls Metriken wie Vollständigkeit, Genauigkeit und Konsistenz, mit denen die Qualität von Metadaten evaluiert werden kann, übertragen lässt [138, 140]. Bei dem Format lässt sich prinzipiell zwischen dem Dateiformat selbst und der Formatierung der eigentlichen Inhalte der Forschungsdaten unterscheiden [139], wobei beide Aspekte in der Regel unabhängig voneinander zu betrachten sind. Beispielsweise können sich zwei unterschiedliche, textbasierte Dateien der Formate JSON oder XML in der Organisation derer Inhalte stark unterscheiden, selbst wenn beide Formate jeweils einer definierten Syntax unterliegen. Während die Dateninhalte bereits durch Dokumentationen und Metadaten ausreichend beschrieben werden können, ist bezüglich des Formats der Daten insbesondere darauf zu achten, dass es sich um möglichst standardisierte, offene und dementsprechend auch langfristig interoperable Formate handelt. Dieser Aspekt kann prinzipiell durch entsprechende FDM-Software forciert werden, was zusätzliche Möglichkeiten wie die automatisierte Extraktion von Metadaten oder die Aufbereitung der Daten in geeignete Visualisierungen ermöglicht, um dadurch wiederum die Auffindbarkeit der Daten verbessern zu können [139]. Ein konkretes Beispiel für die computergestützten Materialwissenschaften stellt das bereits in Kapitel 3 kurz vorgestellte Repositorium NOMAD dar. Wie auch bei den Metadaten, ist letztendlich die Wahl konkreter Formate stark von der entsprechenden Forschungsdisziplin sowie der angestrebten Zielgruppe abhängig [139], weshalb eine Beschränkung von Dateiformaten nur für fachspezifische Anwendungsfälle möglich ist. Weiterhin garantiert die Validierung von Datenformaten, sowie potenziell der Struktur der Dateninhalte, nur eines von vielen Qualitätsmerkmalen, weshalb der Fokus des konzipierten Systems vermehrt auf entsprechende, deskriptive Metadaten gelegt werden sollte.

4.3.2 Datenpersistierung

Auf der technischen Ebene der Persistierung und Verwaltung von Forschungsdaten bietet sich anhand der webbasierten Implementierung des konzipierten Systems

ein sogenannter Objektspeicher (englisch: Object Storage) an [141]. Bei Objektspeichern handelt es sich um ein bereits seit längerem bestehendes Konzept, das durch Cloud-Speichersysteme wie Amazon S3 [142] oder Dropbox [143] in einem weiten Anwendungsgebiet Einsatz findet, jedoch ebenfalls in ähnlicher Form in den meisten Repositorien eingesetzt wird. Das Grundprinzip des Konzepts liegt, im Vergleich zu klassischen Dateisystemen, in der Abstraktion des eigentlichen, zugrunde liegenden Datenspeichers sowie der internen Organisation der Daten, die üblicherweise komplett vom Objektspeicher übernommen wird. Weiterhin wird eine systemunabhängige Autorisierung zum Zugriff auf Daten möglich, die für das konzipierte System bei der Implementierung von Zugriffsbeschränkungen relevant sein kann. Objektspeicher stellen somit weniger einen Ersatz, sondern eine Erweiterung klassischer Speichersysteme dar. Dateien werden jeweils als generische Objekte repräsentiert, die über eindeutige Identifikatoren referenzierbar sind. Bei webbasierten Objektspeichern handelt es sich dabei typischerweise um URLs (Uniform Resource Locator), deren entsprechende Daten über HTTP-APIs manipulierbar sind, wodurch Automatisierung und eine systemunabhängige Integration in existierende Software ermöglicht wird. Ein weiterer Vorteil besteht in der Verwaltung von Metadaten, da Objekte ebenfalls als entsprechende Container von Daten, Metadaten und weiteren Attributen betrachtet werden können, anstatt lediglich als Dateien im engeren Sinne. Dieser Aspekt ist auch mit dem erläuterten Metadatenkonzept vereinbar, welches eine von den Forschungsdaten separate Speicherung von Metadaten und entsprechende Verlinkungen vorsieht.

Potenzielle Nachteile in der Nutzung webbasierter Objektspeicher liegen vor allem in der Verwaltung und Nutzung größerer Datenvolumen. Unter der Annahme, dass ein Objektspeicher als primäres Speichermedium verwendet wird, ist das komplette oder stückweise Herunterladen der Daten notwendig, bevor diese zu einem späteren Zeitpunkt weiterverwendet werden können. Der dabei entstehende Mehraufwand ist für performancekritische Anwendungen signifikant und nach aktuellem Stand kein vollwertiger Ersatz im Vergleich zu spezialisierten, parallelen Dateisystemen [144, 145].

4.3.3 Zusammenfassung

Um die Qualität der mithilfe des konzipierten Systems verwalteten Daten gewährleisten zu können, liegt der Fokus insbesondere auf der flexiblen Spezifikation geeigneter Metadaten, die durch das etablierte Metadatenkonzept bereits abgedeckt ist. Dieser Aspekt ist unabhängig davon wichtig, ob Daten lediglich innerhalb einer bestimmten Arbeitsgruppe genutzt werden oder z. B. über Repositorien der Öffentlichkeit bereitgestellt werden. Letzterer Anwendungsfall kann dennoch weitere Anforderungen beinhalten, wie z. B. die Angabe geeigneter Lizenzen zur Nachnutzung der Forschungsdaten, was prinzipiell jedoch auch für Metadaten gelten kann und entsprechend im konzipierten System berücksichtigt werden muss. Im Hinblick auf Datenformate ist aufgrund der generischen Konzeption des Systems und der heterogenen Bedarfe der Ingenieurwissenschaften eine Beschränkung auf bestimmte Formate nicht sinnvoll, weshalb die Nutzung möglichst standardisierter und etablierter Formate in der Verantwortung der Forscher liegt. Die Extraktion von Metadaten oder Aufbereitung der Forschungsdaten in geeignete Visualisierungen kann für vereinzelte Datenformate dennoch in Betracht gezogen werden.

Zur Speicherung der Forschungsdaten bietet sich, trotz potenzieller Nachteile, die Verwendung eines Objektspeichers an. Dies ist insbesondere durch dessen Flexibilität, Abstraktionsebene und den Möglichkeiten zur Autorisierung begründet, auf technischer Ebene jedoch auch durch die webbasierte Implementierung des konzipierten Systems sowie der vorzugsweisen separaten Speicherung von Forschungsdaten und entsprechender Metadaten. Die Wahl des zugrunde liegenden Datenspeichers spielt dabei hauptsächlich für die potenzielle Anbindung existierender Speichersysteme eine Rolle sowie für die Administration des konzipierten Systems. Im Hinblick auf die Verwaltung größerer Datenvolumen sollte weiterhin die Frage berücksichtigt werden, welche Forschungsdaten überhaupt innerhalb des konzipierten Systems gespeichert werden sollen. Beispielsweise kann unter der vollständigen Angabe von einer zur Generierung von Daten verwendeten Software, entsprechenden Eingabeparametern und -daten sowie passender Metadaten die Reproduzierbarkeit eines Datensatzes ermöglicht werden, ohne die

erzeugten Daten selbst abspeichern zu müssen. Dieser Aspekt erfordert eine anwendungsspezifische Abwägung zwischen den praktischen Anforderungen an die Datenspeicherung und der für die Reproduzierbarkeit solcher Daten erforderlichen Rechenleistung und Umgebung. Eine weitere Möglichkeit besteht in der Verwaltung einfacher Verlinkungen, die auf in externen Speichern oder Systemen persistierte Forschungsdaten verweisen können, in welchem Fall das konzipierte System lediglich die Verknüpfung und Verwaltung entsprechender Metadaten übernimmt.

4.4 Planung von Forschungsvorhaben

Was den Forschungsdatenlebenszyklus betrifft, so beginnt jedes Forschungsprojekt mit der Planung aller innerhalb des Projekts relevanten, FDM-bezogenen Aktivitäten. In diesem Kontext rücken zunehmend DMPs in den Fokus, nicht zuletzt aufgrund der bereits erläuterten Anforderungen verschiedener Förderorganisationen. Zur Erstellung von diesen existieren unterschiedliche Werkzeuge, von denen eine Auswahl in Kapitel 3 vorgestellt wurde und deren Integration daher einer separaten Entwicklung als Teil des konzipierten Systems zu bevorzugen ist. Zur Interoperabilität mit anderen Arten von FDM-Software ist vor allem die Maschinenlesbarkeit von DMPs eine wichtige Voraussetzung. Ähnlich wie bei Forschungsdaten oder Metadaten kann diese z. B. durch die Verwendung von etablierten Datenformaten und standardisierter Terminologie ermöglicht werden [146]. Zwar existieren bereits prototypische Umsetzungen derartiger Ansätze, typischerweise liegt der Fokus von DMPs jedoch in der Verwendung von Freitextbeschreibungen, die sich in Form statischer Dokumente, z. B. im PDF-Format, exportieren lassen [147].

Dennoch ist bereits dadurch eine einfache Form der Integration durch Bereitstellung des entsprechenden Dokuments im konzipierten System denkbar, ohne die Notwendigkeit spezifischer Schnittstellen. Durch die Verknüpfung von im Laufe eines Forschungsvorhabens entstehender Daten und Metadaten, die ebenfalls mithilfe des konzipierten Systems verwalten werden, können konkrete Ergebnisse

mit dem zugrunde liegenden Plan assoziiert werden, was z. B. der konstanten Evaluierung der im Plan dokumentierten Maßnahmen dienen kann. Weiterhin kann der so hinterlegte DMP mit zusätzlichen Metadaten ausgestattet werden, um zumindest allgemeine Informationen in einer interoperablen Art und Weise zu Verfügung stellen zu können.

4.5 Datenerhebung und -aufbereitung

Wie bereits in der Zielsetzung des konzipierten Systems erläutert, spielt die Erhebung und Aufbereitung von Forschungsdaten die wahrscheinlich wichtigste Rolle im Forschungsdatenlebenszyklus. Im Hinblick auf die Datenerhebung fallen bei Betrachtung existierender FDM-Software hauptsächlich ELNs unter diesen Aufgabenbereich. Da diese typischerweise in rein experimentellen und fachspezifischen Arbeitsabläufen zum Einsatz kommen, ist eine Integration existierender ELNs für den Einsatz in den Ingenieurwissenschaften als Teil des konzipierten Systems weniger sinnvoll. Weiterhin sollte die Datenerhebung soweit wie möglich mit den Konzepten für die Verwaltung von Daten und entsprechender Metadaten übereinstimmen, was vor allem eine stärkere Ausrichtung auf die strukturierte und konsistente Datenverwaltung bedeutet. Dennoch können klassische ELN-Funktionalitäten, wie z. B. die Freitextbeschreibung digitalisierter Forschungsprozesse oder die skizzenhafte Dokumentation experimenteller Versuchsaufbauten, berücksichtigt werden.

Da ELNs zunehmend in Bereichen eingesetzt werden, die über den einfachen Ersatz klassischer Laborbücher hinausgehen, lassen sich diesen ebenfalls weitere Aufgabenbereiche zuordnen. Einer dieser Trends liegt in der Laborautomatisierung, die je nach den Anforderungen individueller Labore in unterschiedlichen Formen und Graden umgesetzt werden kann [148]. Neben dem Potenzial zur Verbesserung der Effizienz existierender Arbeitsflüsse, kann dieser Aspekt vor allem zur automatisierten Akquise von Daten und Metadaten beitragen [149], und damit ebenfalls zu deren Qualität. Zwar ist generell eine vollständige Laborautomatisierung aufgrund der Notwendigkeit manuell durchzuführender Schritte

sowie der Diversität verwendeter Gerätschaften selten möglich [51], dennoch bietet vor allem die softwaregestützte Kontrolle entsprechender Geräte Potenzial zur Automatisierung [150]. Für das konzipierte System können dabei unterschiedliche Anforderungen relevant sein, abhängig von der Art und Weise der Datenspeicherung sowie dem Netzwerkzugang von Gerätschaften. Aus technischer Sicht bietet sich insbesondere für diesen Aspekt die Nutzung der ebenfalls bereits in der Zielsetzung des konzipierten Systems angesprochenen HTTP-API an. Diese bietet prinzipiell den Vorteil, von der konkreten Art und Weise der Datenspeicherung eines Geräts unabhängig zu sein, sofern die Möglichkeit besteht, erzeugte Daten und Metadaten über das Netzwerk in Form passender Anfragen zu versenden. Solche Funktionalität kann entweder direkt von einem Gerät oder mithilfe separater Werkzeuge unterstützt werden, die als Brücke zwischen dem Gerät und der HTTP-API des konzipierten Systems dienen können.

Neben den bisher erläuterten experimentellen Arbeitsabläufen sind auch rechnergestützte Arbeitsflüsse relevant, z. B. die Durchführung von Simulationen oder die Analyse und Aufbereitung existierender Forschungsdaten. Mithilfe der durch das konzipierte System bereitgestellten HTTP-API ist die Integration solcher Anwendungsfälle ebenfalls möglich, wobei auch hier unterschiedliche Werkzeuge als Brücke zwischen existierender Software und der HTTP-API dienen können. Die Nutzung der HTTP-API sollte weiterhin einen bidirektionalen Datenfluss ermöglichen, sodass entsprechende Aufbereitungsschritte auch anhand bereits mithilfe des konzipierten Systems verwalteter Forschungsdaten und Metadaten zu einem späteren Zeitpunkt durchgeführt werden können.

4.6 Publizierung, Archivierung und Nachnutzung von Forschungsdaten

Um die langfristige Nutzbarkeit von Forschungsdaten ermöglichen zu können, sind sowohl deren Publizierung als auch Archivierung relevant, wobei insbesondere erstere auf die Nachnutzung der Daten von Forschern externer Institutionen abzielt.

Zur Umsetzung beider Aspekte können unterschiedliche Systeme zum Einsatz kommen. Bei der Archivierung handelt es sich bei diesen typischerweise um institutionelle Speichersysteme, die auf die Langzeitarchivierung von Forschungsdaten ausgelegt sind, während bei der Publizierung in der Regel öffentlich verfügbare Repositorien zu bevorzugen sind, jedoch sind prinzipiell auch Kombinationen beider Lösungen möglich. Aufgrund der heterogenen Natur institutioneller Speichersysteme und entsprechender Anforderungen unterschiedlicher Archivierungskonzepte, ist eine direkte Integration im konzipierten System nicht ohne weiteres möglich. Um diese dennoch unterstützen zu können, bietet sich die Verwendung von Containerformaten an, um Forschungsdaten und entsprechende Metadaten individuell oder als Sammlungen innerhalb geeigneter Datenarchive zu bündeln. Solche Formate können dabei unterstützen, unterschiedliche Daten und Metadaten in einer systemunabhängigen, standardisierten und dadurch maschinenlesbaren Struktur bereitzustellen. Konkrete Beispiele entsprechender Containerformate stellen z. B. RO-Crate oder BagIt dar, die bereits in Kapitel 2 erwähnt wurden. Bei beiden Formaten handelt es sich prinzipiell um einfache Archive regulärer Ordnerstrukturen, die sich z. B. im ZIP-Format serialisieren lassen und deren Aufbau und Inhalte durch in den jeweiligen Standards festgelegte Metadatendateien definiert werden. Entsprechende Import- und Exportmöglichkeiten geeigneter Containerformate sollten daher im konzipierten System berücksichtigt werden, auch da sich Formate wie RO-Crate als eine unkomplizierte und praxisorientierte Implementierung von FDOs betrachten lassen können [151, 152].

Neben der Archivierung können die erläuterten Containerformate ebenfalls bei der Publizierung von Forschungsdaten relevant sein, insbesondere bei der Integration bereits existierender Repositorien. Diese wird für das konzipierte System aufgrund der besonderen Anforderungen an die langfristige Verfügbarkeit publizierter Daten und die Vergabe von PIDs einer separaten Entwicklung vorgezogen. Derartige Aspekte erfordern nicht nur eine entsprechende softwaretechnische Umsetzung, etwa zur Registrierung von DOIs, deren Verwendung sich mittlerweile auch bei Datenpublikationen etabliert hat, sondern auch eine langfristig gesicherte Bereitstellung von Hardwareressourcen zur Speicherung und Bereitstellung der Daten. Um eine möglichst direkte Publizierung von Forschungsdaten und entsprechender

Metadaten innerhalb des konzipierten Systems zu ermöglichen, müssen daher passende Schnittstellen bereitgestellt werden, die eine Integration sowohl generischer als auch fachspezifischer, webbasierter Repositorien ermöglichen. Je nach Repositorium können dabei unterschiedliche Anforderungen an Datenformate und Metadatenstrukturen gestellt werden, wobei durch die Verwendung eines interoperablen Basisschemas im konzipierten System und durch den Export von Containerformaten bereits einige Grundvoraussetzungen gegeben sind. Aspekte wie die Qualität publizierter Forschungsdaten und Metadaten sind dagegen stark anwendungsspezifisch, werden generell jedoch ebenfalls bereits im konzipierten System berücksichtigt.

4.7 Benutzerdefinierte Arbeitsabläufe

Einer der Hauptnachteile der derzeitigen Ausrichtung des konzipierten Systems auf eine generische Entwicklung ist die Abbildung von Arbeitsabläufen, die auf bestimmte Forschungsdisziplinen oder Arbeitsweisen spezialisiert sind. Durch die Bereitstellung einer HTTP-API, die zur Integration existierender Werkzeuge und Gerätschaften dienen kann, lässt sich eine Vielzahl benutzerdefinierter Arbeitsabläufe prinzipiell bereits abdecken. Dennoch können spezifische Anforderungen gegeben sein, die eine Modifikation des konzipierten Systems erfordern, beispielsweise zur direkteren Anbindung bestimmter Speichersysteme oder unterschiedlicher Arten von FDM-Software.

Ein Mechanismus, der in verschiedenen Softwaresystemen häufig zu diesem Zweck eingesetzt wird, ist die Bereitstellung von Pluginschnittstellen zur Erweiterung oder Modifikation verschiedener Funktionalitäten, ohne dass der Kern der Hauptanwendung dazu angepasst werden muss. Entsprechende Plugins können konzeptuell sowohl auf individueller Benutzerbasis, als auch zentral für alle Benutzer einer Software bereitgestellt werden, wobei aufgrund der webbasierten Entwicklung des konzipierten Systems insbesondere letzterer Anwendungsfall relevant ist. Unabhängig davon erfordert die Entwicklung von Plugins das Schreiben von entsprechendem Code, was eine vergleichsweise hohe Hürde für deren

Umsetzung darstellen kann. Daher sollten für einfache Arten von Anpassungen des konzipierten Systems ebenfalls verschiedene Möglichkeiten zur Konfiguration oder Personalisierung auf unterschiedlichen Ebenen zur Verfügung gestellt werden.

4.8 Authentifizierungs- und Autorisierungsinfrastruktur

Wie bereits als Teil der Zielsetzung des konzipierten Systems erläutert, können Zugriffsbeschränkungen für die verwalteten Forschungsdaten und Metadaten relevant sein. Dies spielt ebenfalls innerhalb der FAIR-Prinzipien eine Rolle, welche die Spezifikation von Bedingungen zum Zugriff auf Datensätze erlauben, ohne deren FAIRness zu beeinträchtigen [34]. Gleichzeitig sind Zugriffsbeschränkungen ebenfalls notwendig, wenn Forschungsdaten lediglich innerhalb eines bestimmten Benutzerkreises unter Verwendung des konzipierten Systems geteilt werden sollen. Diese Aspekte erfordern die Umsetzung einer geeigneten Authentifizierungs- und Autorisierungsinfrastruktur (AAI), um die Echtheit von Identitäten bestätigen sowie deren Berechtigungen innerhalb des Systems verifizieren zu können.

Die Authentifizierung von Benutzern stellt generell eine gängige Voraussetzung unterschiedlicher FDM-Software dar, um verwaltete oder publizierte Forschungsdaten den zuständigen Forschern zuordnen zu können, Speicherkontingente zu ermitteln oder den eng damit verbundenen Aspekt der Autorisierung zu ermöglichen. Hierbei müssen insbesondere die unterschiedlichen Umgebungen berücksichtigt werden, in denen entsprechende Software installiert und genutzt werden kann, weshalb verschiedene, bereits etablierte Authentifizierungsmechanismen für das konzipierte System relevant sein können. Im Forschungsumfeld ist insbesondere die Technologie Shibboleth [153] verbreitet, die auf einer Erweiterung des Standards Security Assertion Markup Language (SAML) [154] basiert, eine Spezifikation und Auszeichnungssprache zum Austausch von Authentifizierungs- und

Autorisierungsinformationen. Mithilfe dieser ermöglicht Shibboleth den Benutzern, unterschiedliche Dienste (sogenannte Service Provider) unter Verwendung eines zentralen Authentifizierungsanbieters (ein sogenannter Identity Provider) zu verwenden, wobei es sich bei diesem in der Regel um die Heimatinstitution des jeweiligen Benutzers handelt. Diese Art von Technologie ist allgemein unter dem Begriff Single Sign-on (SSO) bekannt. Was Shibboleth von anderen SSO-Lösungen abhebt, ist die Verbreitung föderierter Zusammenschlüsse unterschiedlicher Identity Provider, was in Form rechtlicher und technischer Standards deren gemeinsame Nutzung erleichtert. Im deutschen Forschungsumfeld existiert hierfür die DFN-AAI [155] des Vereins zur Förderung eines Deutschen Forschungsnetzes (DFN). Eine alternative SSO-Technologie stellt OpenID Connect (OIDC) [156] dar, ein auf dem Autorisierungsprotokoll OAuth 2.0 [157] basierendes System, das dieses um standardisierte Authentifizierungsschnittstellen erweitert. OIDC wird z. B. von ORCID bereitgestellt, um webbasierten Anwendung die Authentifizierung von Benutzern anhand bestehender ORCID-Konten zu ermöglichen. Darüber hinaus wird OIDC zunehmend als modernere Alternative zu Shibboleth verwendet, allerdings fehlen derzeit noch umfassende Lösungen zur Föderation [158].

Eine ebenfalls verbreitete Lösung stellen auf dem Lightweight Directory Access Protocol (LDAP) [159] basierende Systeme dar. Bei LDAP handelt es sich um ein Netzwerkprotokoll, dass zur Abfrage und Manipulation von Informationen in Verzeichnisdiensten eingesetzt wird und von unterschiedlicher Software, wie z. B. OpenLDAP [160], implementiert wird. Insbesondere Benutzerverzeichnisse werden häufig mit LDAP umgesetzt, mit denen eine entsprechende Authentifizierung erfolgen kann. Diese wiederum kann durch Verwendung geeigneter Schnittstellen in unterschiedlichen Systemen genutzt werden, um ebenfalls SSO zu ermöglichen. Während Shibboleth auf der Ebene gesamter Institutionen oder sogar auf nationaler Ebene eingesetzt werden kann, ist die Integration von LDAP speziell für den Einsatz innerhalb einzelner Arbeitsgruppen relevant, da hier oft entsprechende Systeme bereits im Einsatz sind.

Neben der Authentifizierung kann LDAP ebenfalls unterschiedliche Autorisierungsmechanismen bereitstellen, z. B. unter Verwendung zusätzlicher Benutzerattribute oder Gruppenzugehörigkeiten. Das im konzipierten System verwendete Autorisierungskonzept sollte jedoch möglichst unabhängig von spezifischer Software sein und dennoch eine feingranulare und benutzerdefinierte Autorisierung ermöglichen. Für diese Anforderungen kann z. B. eine rollenbasierte Zugriffskontrolle (englisch: Role Based Access Control, kurz RBAC) zum Einsatz kommen, deren Grundkonzept bereits seit den 1990er-Jahren im Einsatz ist [161]. Anstatt Benutzern direkt Zugriffsrechte auf bestimmte Ressourcen zu geben, werden diese über Rollen abstrahiert, welche wiederum die möglichen Aktionen innerhalb unterschiedlicher Ressourcen enthalten. Zwar liegt der ursprüngliche Fokus des Modells auf der Zugriffskontrolle einzelner Rechner oder Dateien, das Konzept ist jedoch auf beliebige Arten digitaler Ressourcen erweiterbar und flexibel genug, anwendungsspezifische Bedarfe abzudecken. Als Erweiterung zu RBAC existiert ebenfalls die sogenannte attributbasierte Zugriffskontrolle (englisch: Attribute Based Access Control, kurz ABAC), deren Grundkonzept aus der eXtensible Access Control Markup Language (XACML) [162] hervorgeht. Diese verwendet unterschiedliche Attribute von Benutzern, Ressourcen oder dem aktuellen Zugriffskontext um entsprechende Zugriffsrechte zu gewähren, die z. B. dazu verwendet werden können, dem Ersteller oder Eigentümer einer Ressource unabhängig von etwaigen Rollen gesonderte Zugriffsrechte zu erteilen. Entsprechende Attribute können dabei ebenfalls aus externen Quellen wie z. B. LDAP-basierten Systemen stammen.

Insbesondere im Kontext von Dateisystemen ist ebenfalls das Konzept der Access Control Lists (ACLs) [163] anzutreffen. ACLs definieren Listen von Benutzern oder Gruppen, um z. B. die Lese- oder Schreibrechte individueller Dateien steuern zu können, was eine feingranularere Zugriffskontrolle im Vergleich zu beispielsweise klassischen Unix-Dateirechten ermöglicht. Zwar bedeutet die Verwaltung entsprechender Listen bei wachsender Zahl von Benutzern einen zunehmenden Verwaltungsaufwand, die zusätzliche Verwendung von Gruppenberechtigungen bietet jedoch eine zu RBAC vergleichbare Mächtigkeit [164]. Da für das konzipierte System je nach Anwendungsfall unterschiedliche Anforderungen an die

Autorisierung gestellt werden können, ist tendenziell eine Kombination der unterschiedlichen Mechanismen und deren jeweiligen Vorteile zu bevorzugen. Hierbei ist insbesondere eine feingranulare und benutzerdefinierte Zugriffskontrolle wichtig, die auf Ebene einzelner Benutzer oder Benutzergruppen eingesetzt werden kann.

4.9 Eine virtuelle Forschungsumgebung für die Ingenieurwissenschaften

Anhand der in diesem Kapitel evaluierten Konzepte und der darauf basierenden, definierten Anforderungen wird deutlich, dass keine im bisherigen Verlauf der Arbeit vorgestellte FDM-Software diese für ingenieurwissenschaftliche Anwendungsfälle in vollem Umfang abdecken kann. Dennoch existieren etliche Systeme, deren Verwendung sich für ausgewählte Bereiche eignen kann, etwa für die Planung von Forschungsvorhaben oder die Publizierung von Forschungsdaten. Für das konzipierte System stellt eine entsprechende Integration daher eine Grundvoraussetzung dar, während der Fokus des Systems selbst auf der Umsetzung bestimmter Kernfunktionalitäten liegt. Diese umfassen die strukturierte Verwaltung von Daten und Metadaten, deren Erhebung unter Verwendung unterschiedlicher Schnittstellen, sowie die zur Umsetzung dieser Funktionalitäten notwendige Infrastruktur. Konzeptionell ist dabei vor allem der Aspekt der Metadatenverwaltung stark an unterschiedlichen, etablierten Konzepten orientiert, um je nach Anwendungsbedarf und Möglichkeiten eine interoperable Strukturierung von diesen gewährleisten zu können, auch wenn kein existierendes Metadatenschema oder -verwaltungssystem direkt zum Einsatz kommt. Die daraus resultierende Kombination aus einer neuartigen Entwicklung und der Adaption bzw. Integration bestehender Konzepte und Software lässt sich am besten als virtuelle Forschungsumgebung (VFU) beschreiben. Im Allgemeinen handelt es sich bei VFUs um meist webbasierte Tools, welche der Zusammenarbeit zwischen Forschern zur Unterstützung des gesamten Forschungsprozesses dienen können [165], wobei

der Schwerpunkt der in dieser Arbeit konzipierten VFU auf dem strukturierten FDM liegt.

Eine Übersicht über das Gesamtkonzept der VFU ist in Abbildung 4.1 dargestellt. Diese lässt sich aus Benutzersicht in zwei logische Komponenten aufteilen, die

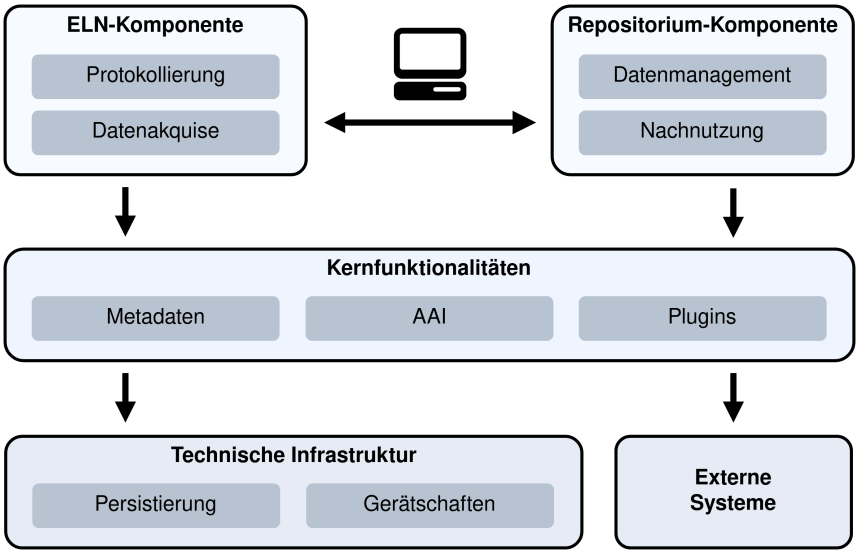

Abbildung 4.1: Gesamtkonzept der konzipierten VFU. Diese gliedert sich logisch in zwei Komponenten, die über gemeinsame Schnittstellen verwendbar sind und jeweils Zugriff auf verschiedene Kernfunktionalitäten haben, sowie indirekt auf die notwendige technische Infrastruktur und externe Systeme.

sich konzeptuell an den Funktionalitäten von ELNs und Repositorien orientieren und über einheitliche Schnittstellen per GUI oder HTTP-API verwendet werden können. Beide Komponenten haben Zugriff auf dieselben Kernfunktionalitäten, wobei Metadatenverwaltung, AAI und Plugininfrastruktur die wichtigsten Aspekte bzw. Module darstellen. Diese wiederum bedienen sich der notwendigen technischen Infrastruktur, die hauptsächlich die Persistierung der Forschungsdaten, Metadaten und weiterer Attribute umfasst, jedoch ebenfalls die Verwendung unterschiedlicher Gerätschaften beinhalten kann. Zuletzt spielt ebenfalls die Integration

externer Systeme eine wichtige Rolle, bei denen es sich insbesondere um die bereits erläuterten Arten von FDM-Systemen handelt.

In der ELN-Komponente liegt der Fokus auf der Protokollierung von Forschungsprozessen sowie auf der Akquise von Forschungsdaten und entsprechender Metadaten. Während erstere hauptsächlich klassische ELN-Funktionalitäten umfasst, z. B. die manuelle Erfassung textueller oder skizzenhafter Beschreibungen, liegt bei der Datenakquise die Automatisierung unter Nutzung der HTTP-API im Vordergrund. Neben der potenziellen Anbindung von Laborgeräten in experimentellen Arbeitsabläufen wird hierdurch ebenfalls die Möglichkeit zur konsistenten Angabe der Datenherkunft eröffnet. Die Repositorium-Komponente dagegen hat die strukturierte Verwaltung und den Austausch der erhobenen Forschungsdaten zur Aufgabe, sowie deren Nachnutzung, welche wiederum die Publizierung und Archivierung der Daten umfassen kann. Trotz der logischen Trennung von Zuständigkeiten beider Komponenten lässt sich das gesamte System als eine Einheit betrachten, bei der sämtliche Module unter Nutzung entsprechender Kernfunktionalitäten und Infrastruktur zusammenspielen.

Durch die Kombination beider Komponenten werden unterschiedliche, ingenieurwissenschaftliche Forschungsprozesse ermöglicht, wobei sowohl experimentelle, simulative als auch hybride Arbeitsflüsse berücksichtigt werden. Da die FAIR-Prinzipien insbesondere auf die Publizierung von Forschungsdaten abzielen, spielen die unterschiedlichen Phasen des Forschungsdatenlebenszyklus ebenfalls eine wichtige Rolle. Dies ermöglicht nicht nur strukturiertes FDM „von Anfang an“, sondern bietet ebenfalls Anknüpfungspunkte für die Integration existierender Arbeitsabläufe und Systeme. Der Schwerpunkt liegt darauf, zu gewährleisten, dass die konzipierte VFU von den Forschern selbst genutzt werden kann, um die schrittweise FAIRness von Forschungsdaten und Metadaten in Form von praxisorientierten, Bottom-up-Ansätzen zu unterstützen. Insbesondere wird hierbei die Verwendung im Kontext des sogenannten Long Tail of Science betrachtet. Hierbei handelt es sich um alle Arten von Forschungsprozessen, unabhängig von der konkreten Disziplin, die z. B. im Rahmen kleinerer Forschungsprojekte mit

begrenzten, finanziellen Ressourcen oder von Forschern und Institutionen durchgeführt werden, denen das notwendige Fachwissen zur Umsetzung eines strukturierten FDMs fehlt. Entsprechende Forschungsdaten werden auch als „dunkle Daten“ bezeichnet [53]. Dieser Aspekt betrifft auch die möglichst unkomplizierte Administration der konzipierten VFU, die ihrerseits jedoch von der konkreten Implementierung der Konzepte abhängt.

5 Ergebnisse

Um die in Kapitel 4 erläuterten Konzepte programmiertechnisch umsetzen und damit in der Praxis anwenden und evaluieren zu können, wird in diesem Kapitel die konkrete Implementierung einer darauf ausgerichteten VFU für die Ingenieurwissenschaften beschrieben. Diese wird zunächst auf der funktionalen Ebene einzelner Komponenten betrachtet und anschließend anhand eines fiktiven Experiments im Kontext eines entsprechenden Arbeitsflusses angewandt. Weiterhin werden die wichtigsten Aspekte der technischen Umsetzung und die damit verbundenen Entwurfsentscheidungen im Hinblick auf die konzeptuellen Grundlagen erläutert.

Bei der implementierten VFU handelt es sich um das System Kadi4Mat (Karlsruher Dateninfrastruktur für die Materialwissenschaften) [166], dessen Logo in Abbildung 5.1 dargestellt ist. Kadi4Mat ist im Rahmen verschiedener Forschungs-

Abbildung 5.1: Logo von Kadi4Mat.

projekte am KIT mit einem anfänglichen Schwerpunkt auf den Materialwissenschaften entstanden. Dieser Schwerpunkt wurde jedoch im Laufe der Entwicklung um allgemeinere, ingenieurwissenschaftliche Anforderungen entsprechend der in Kapitel 4 erläuterten Konzepte erweitert. Da es sich um eine fortlaufende Entwicklung handelt, entspricht der in diesem Kapitel beschriebene funktionale und technische Stand von Kadi4Mat der veröffentlichten Version 0.42.0 [167].

5.1 Überblick über das Gesamtsystem

Bei Kadi4Mat handelt es sich um ein webbasiertes System, bei dem eine klassische Client-Server-Architektur zum Einsatz kommt. Zur Umsetzung der übergreifenden Webinfrastruktur wird die Bibliothek Flask [168] eingesetzt, ein in der Programmiersprache Python implementiertes Webframework. Dieses ist mit dem Web Server Gateway Interface (WSGI) [169] kompatibel, eine Spezifikation für die Programmiersprache Python, die eine standardisierte Schnittstelle zwischen Webservern und Webframeworks festlegt. Als sogenanntes Mikroframework ist die Funktionalität von Flask selbst auf die für Webanwendungen benötigten Grundfunktionen beschränkt und steht damit im Gegensatz zu Frameworks wie Django [170], das sich durch eine Vielzahl bereits mitgelieferter Komponenten, etwa zur Anbindung von Datenbanken, hervorhebt. Zur Ausführung von länger laufenden Hintergrundprozessen kommt die Software Celery [171] zum Einsatz. Diese stellt eine asynchrone Aufgabenwarteschlange (englisch: Task Queue) bereit, sowie entsprechende Infrastruktur zur Verwaltung und Abarbeitung der eigentlichen Aufgaben. Zur Kommunikation zwischen Kadi4Mat und den entsprechenden Arbeiterprozessen, welche die Ausführung der Aufgaben übernehmen, wird ein Nachrichtenbroker (englisch: Message Broker) eingesetzt. Hierfür kommt aktuell das System Redis [172] zum Einsatz, eine In-Memory-Datenbank, die konzeptuell Daten in Form von einfachen Schlüssel-Wert-Paaren speichert und damit der Familie der NoSQL-Datenbanken (Not only SQL) angehört.

Kadi4Mats Funktionalitäten sind über eine einheitliche GUI mithilfe eines regulären Webbrowsers verwendbar, wodurch eine system- und geräteunabhängige Nutzung ermöglicht wird. Für den programmiertechnischen und automatisierten Gebrauch ist ein Großteil der über die GUI verwendbaren Funktionalitäten ebenfalls über eine entsprechende HTTP-API verfügbar. Beide Schnittstellen setzen eine vorherige Authentifizierung der Benutzer voraus, wobei unterschiedliche Mechanismen zur Verfügung gestellt werden. Um die eigentliche Organisation von Forschungsdaten zu ermöglichen, stellt Kadi4Mat unterschiedliche Arten von Ressourcen bereit, die mithilfe der Schnittstellen erstellt, bearbeitet und verwaltet

werden können. Eine entsprechende Übersicht, sowie die Beziehungen der Ressourcen zueinander, ist in Abbildung 5.2 dargestellt. Auch wenn die Ressourcen

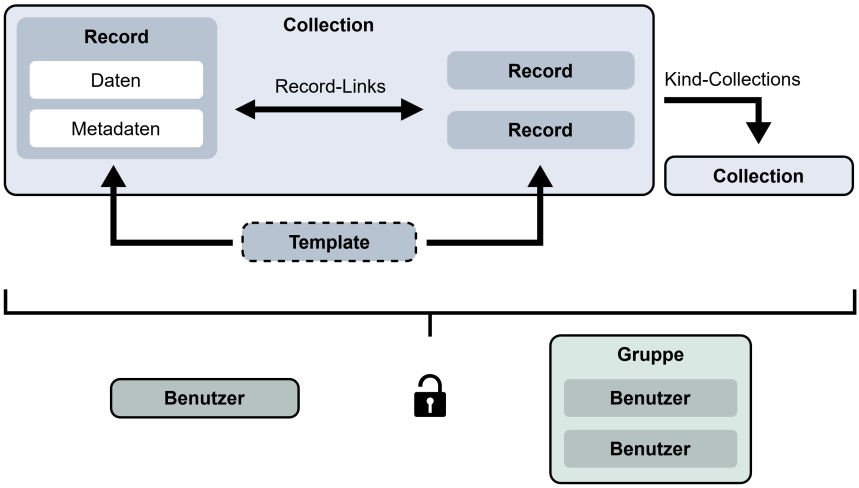

Abbildung 5.2: Unterschiedliche Ressourcen und deren Beziehungen in Kadi4Mat. Records stellen hierbei die Grundkomponente dar und gruppieren Daten mit entsprechenden Metadaten. Templates können zu deren einfacher Erstellung beitragen, während Collections der Organisation mehrerer Records oder Collections dienen können. Sowohl individuellen Benutzern als auch Gruppen mehrerer Benutzer können feingranulare Zugriffsrechte auf die unterschiedlichen Ressourcen erteilt werden.

unterschiedlichen Zwecken dienen, verfügen diese dennoch über einige gemeinsame Funktionalitäten. Diese umfassen die Verwaltung grundlegender Metadaten, Revisionen zur Nachverfolgung von Änderungen der Metadaten und sonstiger Attribute, Exportfunktionalitäten der Ressourceninhalte in verschiedene Formate sowie die Vergabe von feingranularen Zugriffsrechten. Auf einige dieser Aspekte wird im weiteren Verlauf dieses Kapitels ausführlicher eingegangen.

Die wichtigste Ressource sind die sogenannten *Records*, welche die Grundkomponente des FDMs in Kadi4Mat bilden. Aus konzeptueller Sicht repräsentiert jeder Record ein digitales oder digitalisiertes Objekt, das aus beliebigen Forschungsdaten sowie zugehörigen Metadaten bestehen kann, die in Form des Records

innerhalb eines Containers logisch gruppiert werden. Neben grundlegenden Metadaten, die einem fixen Schema folgen, ist ebenfalls die Angabe generischer bzw. anwendungsspezifischer Metadaten möglich. Die Angabe von Daten ist optional, sodass neben regulären Forschungsdaten, wie z. B. Messdaten oder Simulationsergebnissen, ebenfalls die Repräsentation von beispielsweise Laborgeräten oder der Durchführung individueller Forschungsprozesse möglich wird, die üblicherweise lediglich aus Metadaten bestehen. Während individuelle Records komplett eigenständig sein können, kann zur vollständigen Repräsentation der Datenherkunft die Beziehung mehrerer Records zueinander relevant sein. Für diese Anforderung lassen sich mehrere Records durch sogenannte *Record-Links* miteinander verknüpfen. So können z. B. ein Laborgerät, dadurch entstandene Messdaten sowie der zugrunde liegende Prozess jeweils als Record repräsentiert und durch entsprechende Record-Links in Beziehung gesetzt werden.

Zur einfachen Erstellung von Records können *Templates* beitragen, was insbesondere bei Verwendung der GUI von Kadi4Mat eine essenzielle Rolle spielt, um die manuelle Erstellung von Records durch die Vorgabe unterschiedlicher Metadaten, Validierungsanweisungen oder Hilfestellungen zu vereinfachen. Sowohl für gesamte Records, als auch speziell für deren generische Metadaten, lassen sich entsprechende Templates definieren. Templates können somit mehr oder weniger die Rolle von benutzerdefinierten und flexiblen Metadatenschemata übernehmen. Prinzipiell sind die Inhalte von Templates nicht auf Records beschränkt, sondern richten sich nach verschiedenen möglichen Typen, wobei aktuell die Erstellung von Records im Fokus steht.

Während Records durch entsprechende Verlinkungen bereits untereinander in Beziehung gesetzt werden können, ist eine weitere Form der Gruppierung sinnvoll, um zusammengehörende Records, z. B. sämtliche Objekte und Prozesse eines einzelnen Experiments, geeignet organisieren zu können. Hierfür dienen die sogenannten *Collections*, die logische Gruppierungen mehrerer Records repräsentieren, wobei ein einzelner Record prinzipiell auch Teil mehrerer Collections sein kann, unabhängig von den Verlinkungen der Records untereinander. Neben Records lassen sich ebenfalls Kind-Collections definieren, wobei jede Collection maximal

eine übergeordnete Collection besitzen kann. Dadurch wird eine hierarchische Struktur ermöglicht, die z. B. einem typischen Dateisystem ähnelt, bei dem jede Collection einem Ordner entsprechen würde. Im Gegensatz zu Records beinhalten Collections lediglich grundlegende Metadaten.

Gruppen stellen die letzte Art von Ressource in Kadi4Mat dar. Ähnlich wie Collections dienen Gruppen dazu, vorhandene Ressourcen besser organisieren zu können, allerdings speziell im Hinblick auf die Verwaltung von Zugriffsrechten. Gruppen können eine beliebige Anzahl an Mitgliedern besitzen, wobei sämtliche registrierte Benutzer einer konkreten Kadi4Mat-Instanz einbezogen werden können. Wird anschließend einer Gruppe ein bestimmtes Zugriffsrecht erteilt, wird dieses automatisch auf alle Mitglieder der Gruppe angewandt. Durch die benutzerdefinierte Verwaltung von Gruppen und der Kombination mit Zugriffsrechten auf individueller Benutzerebene wird ein flexibler Autorisierungsmechanismus ermöglicht.

5.2 Strukturierte Verwaltung von Metadaten

Die strukturierte Verwaltung von Metadaten stellt eine der wichtigsten Kernfunktionalitäten von Kadi4Mat dar, weshalb dieser Aspekt in den folgenden Abschnitten gesondert betrachtet wird. Insgesamt orientiert sich die entsprechende Implementierung stark an den in Kapitel 4 erläuterten Metadatenkonzepten, angefangen mit der Verwendung eines Basisschemas, welches für die verschiedenen in Kadi4Mat verwendbaren Ressourcen zum Einsatz kommt und als Teil der Records zusätzlich um weitere, generische Metadaten erweiterbar ist.

5.2.1 Basisschema

Das in Kadi4Mat verwendete Basisschema enthält lediglich grundlegende Metadatenelemente, die je nach Art der Ressource teilweise um weitere Einträge

ergänzt werden. Der Einfachheit halber liegt der Schwerpunkt auf der Betrachtung der Record-Metadaten, wobei intern verwendete Metadaten wie der aktuelle Zustand oder die standardmäßige Sichtbarkeit eines Records, die zur Beschreibung des eigentlichen Inhalts nicht relevant sind, vernachlässigt werden. Das entsprechende Basisschema setzt sich aus den in Tabelle 5.1 gelisteten Einträgen zusammen, wobei die meisten Metadatenelemente in gleicher Form auch innerhalb anderer Ressourcentypen verwendet werden. Da es sich bei den ver-

Tabelle 5.1: Grundlegende Metadatenelemente des Basisschemas von Records.

Element	Beschreibung
ID	Automatisch generierter, numerischer und innerhalb einer konkreten Instanz von Kadi4Mat eindeutiger Identifikator eines Records.
Identifier	Benutzerdefinierter und -lesbarer Identifikator eines Records, der ebenfalls innerhalb einer konkreten Instanz von Kadi4Mat eindeutig, im Gegensatz zur ID jedoch änderbar ist.
Titel	Benutzerdefinierter Titel eines Records. Dieser muss im Gegensatz zum Identifier nicht eindeutig sein.
Beschreibung	Benutzerdefinierte und optionale Beschreibung eines Records als Freitext, die Markdown-Syntax zur formatierten Darstellung innerhalb einer GUI unterstützt.
Typ	Benutzerdefinierter und optionaler Typ eines Records, z. B. *Datensatz* oder *Versuchsgerät*.
Tags	Benutzerdefinierte und optionale Schlüsselwörter zur weiteren Kategorisierung eines Records.
Lizenz	Optionale Lizenz eines Records, die aus einer Liste vordefinierter Lizenzen ausgewählt werden kann und beim Teilen oder Publizieren eines Records relevant ist.
Ersteller	Automatisch generierte Referenz auf den Ersteller eines Records.
Zeitstempel	Automatisch generierte Zeitstempel, welche den Zeitpunkt der Erstellung und letzten Änderung eines Records beinhalten.

wendeten Elementen größtenteils um administrative Metadaten handelt, ist ein Crosswalk zu existierenden, generischen Metadatenschemata prinzipiell möglich. Tabelle 5.2 zeigt eine entsprechende Möglichkeit unter Verwendung von Elementen der DCMI Metadata Terms. Trotz des gültigen Crosswalks werden

Tabelle 5.2: Möglicher Crosswalk des Basisschemas von Records zu Elementen der DCMI Metadata Terms. Zur Spezifizierung letzterer wird das gängige Präfix bzw. der Namensraum `dcterms` verwendet.

Name	DCMI Metadata Term	Bemerkung
ID	`dcterms:identifier`	Als Teil einer entsprechenden URL, abhängig von der konkreten Kadi4Mat-Instanz.
Titel	`dcterms:title`	-
Beschreibung	`dcterms:description`	-
Typ	`dcterms:type`	Die Verwendung eines kontrollierten Vokabulars ist empfohlen, jedoch keine Voraussetzung.
Tags	`dcterms:subject`	Analog zum Typ.
Lizenz	`dcterms:license`	Bei standardisierten Lizenzen in Form einer entsprechenden URL, welche die Lizenz beschreibt.
Ersteller	`dcterms:creator`	Entweder die ORCID (falls angegeben), eine URL (siehe ID) oder der Anzeigename eines Benutzers.
Zeitstempel	`dcterms:date`	Erstell- oder Änderungsdatum eines Records, z. B. nach ISO 8601-Standard [173].

bereits hier mögliche Einschränkungen sichtbar, wie z. B. die Verwendung eines kontrollierten Vokabulars für die Spezifikation eines Typs, da Kadi4Mat aktuell keine Beschränkungen für diesen vorgibt. Während innerhalb der DCMI Metadata

Terms die Verwendung des DCMI Type Vocabulary bevorzugt wird, spezifizieren Schemata wie z. B. das DataCite Metadata Schema eine Kombination aus Freitext und eigens definierten Werten für mögliche Ressourcentypen. Weiterhin lassen beide Schemata unterschiedliche Datumsformate zu, was die syntaktische Interoperabilität der Metadaten beeinflussen kann. Dennoch liefert das Basisschema das bereits im Konzept hervorgehobene, grundlegende Maß an Interoperabilität. Unterschiedliche Konvertierungen und mögliche Inkompatibilitäten müssen dabei jeweils im Kontext bestimmter Anwendungsfälle betrachtet werden. Konkrete Beispiele von bereits in Kadi4Mat enthaltenen und in diesem Zusammenhang relevanten Exportfunktionalitäten werden im weiteren Verlauf dieses Kapitels näher betrachtet.

5.2.2 Generische Metadaten

Die Mächtigkeit und Flexibilität von Records liegt in der Spezifikation von benutzerdefinierten und domänenspezifischen Metadaten, die für jeden Record individuell definiert werden können. Diese werden innerhalb von Kadi4Mat als *Extra-Metadaten* oder kurz *Extras* bezeichnet, im weiteren Verlauf der Arbeit sind diese jedoch ebenfalls mit der allgemeinen Bezeichnung *generische Metadaten* gleichzusetzen. Während die generischen Metadaten konzeptuell formatagnostisch sind, werden diese in den folgenden Beispielen in JSON-Syntax angegeben. Dies ist auf die geringe Komplexität des JSON-Formats und die strukturelle Ähnlichkeit zwischen den Grundkomponenten des Formats und den generischen Metadaten zurückzuführen. Weiterhin spielen ebenfalls die im späteren Verlauf des Kapitels erläuterte Persistierung und Indexierung der Metadaten eine Rolle, denen dasselbe Format zugrunde liegt.

Die generischen Metadaten bestehen im Kern aus einer geordneten Sammlung erweiterter Schlüssel-Wert-Paare unterschiedlicher Typen. Abbildung 5.3 zeigt ein einfaches Beispiel eines entsprechenden Metadatums. Dieses besteht aus der Spezifikation eines Datentyps (`type`), eines Schlüssels (`key`) und eines entsprechenden Werts (`value`). Obwohl in diesem Beispiel der Typ auch auf der Grundlage des

```
{
    "type": "str",
    "key": "name",
    "value": "Nico"
}
```

Abbildung 5.3: Beispiel eines einfachen, generischen Record-Metadatums in JSON-Syntax.

Wertes abgeleitet werden könnte, ist er obligatorisch, da die Angabe eines Wertes wiederum optional ist, was z. B. in JSON-Syntax durch den Wert `null` dargestellt werden kann. Eine Übersicht über die verschiedenen Typen ist in Tabelle 5.3 dargestellt. Die Typen orientieren sich insgesamt größtenteils am JSON-Format.

Tabelle 5.3: Übersicht über die verschiedenen Typen der generischen Record-Metadaten.

Typ	Kurzform	Beschreibung
String	`str`	Ein (nicht leerer) textueller Wert.
Integer	`int`	Ein ganzzahliger Wert, der auf Werte zwischen $-(2^{53}-1)$ und $2^{53}-1$ begrenzt ist.
Float	`float`	Eine Gleitkommazahl doppelter Genauigkeit (64 Bit).
Boolean	`bool`	Ein (binärer) boolescher Wert, der z. B. in JSON-Syntax durch `true` oder `false` spezifiziert wird.
Date	`date`	Ein Datums- und Zeitwert, formatiert gemäß dem ISO 8601-Standard.
Dictionary	`dict`	Ein verschachtelter Wert, der verwendet werden kann, um mehrere Metadateneinträge unter einem einzigen Schlüssel zusammenzufassen.
List	`list`	Ein verschachtelter Wert, der ähnlich wie ein Dictionary funktioniert, mit dem Unterschied, dass keiner der Werte in einer Liste einen Schlüssel besitzt.

String- und Boolean-Werte sind unverändert von diesem übernommen, während der allgemein gehaltene, numerische Typ in JSON (*Number*) in zwei separate

Typen für Ganzzahlen und Gleitkommazahlen aufgeteilt wird, deren Wertebereiche wiederum auf technischen Einschränkungen beruhen. Ein gesonderter Typ für Datums- und Zeitwerte existiert in JSON nicht, wird der Einfachheit halber jedoch als separater Typ in Kadi4Mat bereitgestellt. Mithilfe der letzten zwei in Tabelle 5.3 gelisteten Typen ist es dagegen möglich, beliebig komplexe Metadatenstrukturen zu definieren. Diese entsprechen den im JSON-Format vorhandenen *Objekten* (*Dictionary* in Kadi4Mat) und *Arrays* (*List* in Kadi4Mat), welche zur Repräsentation von verschachtelten Werten unterschiedlicher Typen verwendet werden können. Ersterer Typ kommt genau genommen bereits in Abbildung 5.3 zur Gruppierung der einzelnen Bestandteile eines Metadatums zum Einsatz, dient hier jedoch lediglich syntaktischen Zwecken.

Abbildung 5.4 zeigt ein erweitertes Beispiel unter Verwendung eines verschachtelten Typs, mit dem verschiedene Eigenschaften einer Person beschrieben werden. Als Container dient ein Dictionary-Wert mit dem Schlüssel `person`, der zwei

```
{
    "type": "dict",
    "key": "person",
    "value": [
        {
            "type": "str",
            "key": "name",
            "value": "Nico"
        },
        {
            "type": "int",
            "key": "age",
            "value": 28,
            "unit": null
        }
    ]
}
```

Abbildung 5.4: Beispiel verschachtelter, generischer Record-Metadaten in JSON-Syntax.

Metadaten einfachen Typs enthält, welche den Namen und das Alter der Person beschreiben. Beide Metadaten sind zusätzlich von einem Array umgeben, welches diesen eine definierte Reihenfolge zuweist, während gleichzeitig die Struktur einzelner Metadatenelemente konsistent bleiben kann. Bei näherer Betrachtung des in Abbildung 5.4 dargestellten Integer-Werts fällt weiterhin die Verwendung eines zusätzlichen Attributs `unit` auf. Dieses ermöglicht die optionale Angabe einer textuellen Einheit, welche den entsprechenden Wert näher beschreibt und aktuell für beide Arten numerischer Werte unterstützt wird.

Mithilfe der beschriebenen Funktionalität ist bereits eine komplett generische und benutzer- bzw. anwendungsspezifische Definition von Metadaten möglich. Der Nachteil bei der Angabe von Metadaten, wie sie in Abbildung 5.4 dargestellt sind, liegt in der fehlenden Interoperabilität mit existierenden Metadatenschemata oder, allgemeiner betrachtet, kontrollierten Vokabularen. Um diese zu unterstützen, lassen sich Terme für individuelle Metadaten in Form von IRIs spezifizieren, deren zugrunde liegenden Konzepte das jeweilige Metadatum repräsentieren können. Abbildung 5.5 zeigt eine entsprechende Erweiterung der bereits in Abbildung 5.4 dargestellten Metadaten. Bei den jeweils durch das `term`-Attribut angegebenen Termen handelt es sich um die konzeptuelle Beschreibung einer Person (`person`) bzw. deren Vorname (`name`) gemäß der Definition des Schema.org-Vokabulars. Hierbei besteht keine Beschränkung in Bezug auf die Nutzung bestimmter Vokabulare. Ebenfalls ist die Kombination mit eigens definierten Metadaten möglich, wie z. B. die Angabe des Alters (`age`) der in Abbildung 5.4 beschriebenen Person. Neben der Beschreibung von Konzepten lassen sich prinzipiell auch weitere Informationen als Teil der Terme hinterlegen, die häufig über den entsprechenden IRI (bei Verwendung von diesem als URL) abrufbar sind. Bei diesen kann es sich z. B. um zusätzliche Beschreibungen, nutzerlesbare Titel unterschiedlicher Sprachen, Synonyme oder standardisierte Einheiten handeln, die aktuell nicht in dieser Form innerhalb der generischen Metadaten in Kadi4Mat spezifizierbar sind. Die generischen Metadaten lassen sich daher als Vorstufe bzw. Referenz entsprechend standardisierter Konzepte betrachten und enthalten die dafür wichtigsten

```
{
    "type": "dict",
    "key": "person",
    "term": "https://schema.org/Person",
    "value": [
        {
            "type": "str",
            "key": "name",
            "value": "Nico",
            "term": "https://schema.org/givenName"
        },
        {
            "type": "int",
            "key": "age",
            "value": 28,
            "unit": null
        }
    ]
}
```

Abbildung 5.5: Beispiel generischer Record-Metadaten unter Verwendung existierender Terme bzw. Konzepte des Schema.org-Vokabulars in JSON-Syntax.

Attribute in Form von Schlüsseln, Werten verschiedener Typen und einfachen Einheiten, wobei letztere ebenfalls in Form eines separaten, textuellen Metadatums mit optionalem Term angegeben werden können.

Zuletzt ist ebenfalls die Spezifikation optionaler Validierungsanweisungen innerhalb der generischen Metadaten möglich, die hauptsächlich bei der Definition und Nutzung von Templates in Kadi4Mat zum Einsatz kommen. Diese ähneln den vergleichbaren Funktionalitäten des Standards JSON Schema [174], allerdings handelt es sich um eine stark vereinfachte Umsetzung, um die Komplexität der Definition entsprechender Validierungsanweisungen gering zu halten und die Kompatibilität mit der bereits bestehenden Metadatenstruktur zu gewährleisten. Abbildung 5.6 zeigt ein Beispiel eines entsprechenden Metadatums. Die Validierungsanweisungen werden durch das `validation`-Attribut spezifiziert und

```
{
    "type": "str",
    "key": "type",
    "value": null,
    "validation": {
        "required": true,
        "options": ["dataset", "device", "software"]
    }
}
```

Abbildung 5.6: Beispiel eines generischen Record-Metadatums unter Verwendung optionaler Validierungsanweisungen in JSON-Syntax.

dienen in diesem Beispiel dazu, die Angabe eines Wertes erforderlich zu machen (`required`) und gleichzeitig die möglichen, konkreten Werte einzuschränken (`options`). Als Teil eines konkreten Records wäre das in Abbildung 5.6 dargestellte Beispiel daher nicht valide, allerdings als Teil eines Templates, dessen konkrete Werte auch zu einem späteren Zeitpunkt definiert werden können. Neben diesen beiden Möglichkeiten lassen sich ebenfalls mögliche Wertebereiche (`range`) für numerische Werte definieren, womit bereits die wichtigsten, grundlegenden Validierungen unterstützt werden. Während einige dieser Aspekte prinzipiell ebenfalls als Teil eines Terms bzw. dem zugrunde liegenden Konzept spezifiziert werden können, wird hierdurch insbesondere die manuelle Erfassung von Metadaten mithilfe der GUI von Kadi4Mat erleichtert.

5.2.3 Verlinkungen und Datenherkunft

Wie zu Beginn dieses Kapitels bereits erwähnt, können Record-Links dazu dienen, die Beziehungen mehrerer Records untereinander zu spezifizieren, was in Kombination mit der Angabe generischer Metadaten insbesondere dazu dient, die vollständige Herkunft von Daten erfassen zu können. Jeder Link verbindet dabei immer genau zwei Records in einer definierten Richtung miteinander, weshalb diese auch als spezielle Form von Metadaten der verlinkten Records betrachtet werden können. Record-Links stellen konzeptuell jedoch eine separate Art von

Ressource dar und enthalten selbst wiederum Metadaten. Diese umfassen insbesondere den benutzerdefinierten Namen des Links bzw. die Art der Relation beider Records zueinander, jedoch werden ebenfalls grundlegende Metadaten entsprechend des Basisschemas verwendet, wie die eindeutige ID des Links, Referenzen auf den Ersteller eines Links und verschiedene Zeitstempel. Letztere werden außerdem in Kombination mit entsprechenden Revisionen der verlinkten Records dazu verwendet, potenzielle Änderungen seit dem Zeitpunkt der Verlinkung einsehen zu können. Weiterhin ist auch für individuelle Record-Links die Angabe eines Terms in Form eines entsprechenden IRIs möglich, um ähnlich zu den generischen Metadaten ein dem Link zugrunde liegendes Konzept referenzieren zu können.

Neben der Spezifikation der Datenherkunft besteht ein weiterer Anwendungsfall in der Verknüpfung von einem in Form eines Records gespeicherten DMPs mit den assoziierten Forschungsdaten und Metadaten weiterer Records, wodurch die in Kapitel 4 erläuterte, einfache Integration von DMP-Software möglich wird. Record-Links können ebenfalls bei Crosswalks zu Ontologien wie PROV eingesetzt werden. Während individuelle Records verschiedene Entitäten und Aktivitäten in PROV repräsentieren können, werden diese mithilfe entsprechender Record-Links in Relation zueinander gesetzt und durch die zugehörigen Metadaten genauer beschrieben. Bei den Agenten, welche den Entitäten und Aktivitäten zugeordnet sind, kann es sich sowohl um die Ersteller der unterschiedlichen Records handeln, die innerhalb von Kadi4Mat automatisch hinterlegt werden, oder um separate Records, welche in diesem Fall ebenfalls Software oder Gerätschaften repräsentieren können. Ähnlich wie bei möglichen Crosswalks des Basisschemas handelt es sich daher um eine stark anwendungsspezifische Konvertierung.

5.2.4 Persistierung

In Kadi4Mat kommt das relationale Datenbankmanagementsystem (englisch: Relational Database Management System, kurz RDBMS) PostgreSQL [175] zur Persistierung sämtlicher Metadaten und weiterer Aspekte, wie der Verwaltung von Benutzerkonten oder Zugriffsrechten, zum Einsatz. Bei relationalen Datenbanken

handelt es sich um digitale Datenbestände, deren Struktur auf dem von Codd vorgeschlagenen relationalen Datenmodell [176] aufbaut. Die Grundlage solcher Systeme besteht darin, persistierte Daten in Form von Tabellen bzw. Relationen zu repräsentieren. Jede Tabelle gibt eine unterschiedliche Anzahl Spalten verschiedener Typen vor, die durch ein zuvor definiertes Datenbankschema festgelegt werden. Die Zeilen jeder Tabelle stellen wiederum die eigentlichen Datensätze dar. Zur Abfrage und Manipulation der Datensätze wird typischerweise die Abfragesprache SQL (englisch: Structured Query Language) eingesetzt. In Kadi4Mat wird PostgreSQL in Kombination mit einem entsprechenden Object-Relational Mapping (ORM) verwendet. Mithilfe von ORMs ist es möglich, Daten zwischen typischen Konstrukten verschiedener Programmiersprachen und innerhalb einer Datenbank persistierten Relationen zu konvertieren. Die zugrunde liegenden SQL-Abfragen zur Manipulation der Daten werden vom ORM automatisch generiert, wodurch das Programmiermodell existierender Software nicht angepasst werden muss. Konkret wird die Python-Bibliothek SQLAlchemy [177] verwendet, die eine datenbankagnostische und objektorientierte Definition von Relationen sowie die Abfrage und Manipulation über entsprechende, im Python-Code verwaltete Klassen und Objekte unterstützt.

Die Wahl von PostgreSQL als zugrunde liegendes RDBMS ist neben dem allgemeinen Reifegrad der Software auch in der Unterstützung von nicht-relationalen Datentypen wie z. B. JSON begründet. Dies wird u. a. mithilfe des nativen JSONB-Datentyps ermöglicht, welcher die effiziente Speicherung von JSON-Daten in einem dafür vorgesehenen Binärformat sowie die Abfrage einzelner Schlüssel oder Werte auf Datenbankebene ermöglicht. Dadurch können die generischen Metadaten von Records direkt in der JSON-basierten Metadatenstruktur persistiert werden, die z. B. in Abbildung 5.5 dargestellt wird. Dies ermöglicht die einfache Kombination aus der Verwendung etablierter Standards wie SQL und semistrukturierter Daten, deren Flexibilität ansonsten üblicherweise nur in NoSQL-Datenbanken zu finden ist.

Eine potenzielle Alternative stellt das Entity-Attribute-Value-Modell (EAV) dar. Die Grundidee dieses Modells besteht darin, Entitäten (englisch: Entity), deren

Attribute (englisch: Attribute) und Werte (englisch: Value) innerhalb einer flachen Hierarchie so zu beschreiben, dass keine kontinuierlichen Änderungen des Datenbankschemas notwendig werden. Eine mögliche Umsetzung dieses Ansatzes unter Verwendung der beispielhaften Metadaten von Abbildung 5.4 ist in Tabelle 5.4 dargestellt. Für jede Entität, in diesem Fall die durch einen Schlüssel

Tabelle 5.4: Beispiel persistierter Metadaten unter Verwendung eines typischen EAV-Modells.

key	type	str_value	int_value
"person"	"dict"	NULL	NULL
"person.name"	"str"	"Nico"	NULL
"person.age"	"int"	NULL	28

beschriebenen Metadatenelemente, wird deren Typ und optional ein diesem Typ entsprechender Wert gespeichert, wobei für verschachtelte Werte die Notation `schlüssel_1.schlüssel_2` verwendet wird. Pro Typ kommt eine separate und passend typisierte Spalte innerhalb der Datenbank zum Einsatz. Dieses Modell lässt sich um zusätzlichen Typen und Spalten erweitern, wie z. B. auch für die Persistierung von optionalen Einheiten, sofern die Gesamtzahl an Spalten begrenzt bleibt. Während mit dem EAV-Modell die Einschränkungen der Struktur relationaler Datenbanken umgangen werden können, zeigen sich jedoch neben der geringeren Komplexität bei der Verwendung des JSONB-Datentyps ebenfalls Vorteile im Hinblick auf den benötigten Speicherbedarf und der Abfragezeit von Attributen [105].

5.2.5 Indexierung und Suche

Wie innerhalb der in Kapitel 4 erläuterten Zielsetzung des konzipierten Systems erwähnt, kann die Auffindbarkeit von Forschungsdaten und Metadaten über entsprechende Suchmaschinen ermöglicht werden. Auch wenn dieser Aspekt insbesondere auf die Wiederverwendbarkeit publizierter Forschungsdaten abzielt,

ist eine entsprechende Funktionalität auch für Kadi4Mat selbst relevant, um die strukturierte Organisation von Forschungsdaten zu unterstützen. Dies setzt neben der existierenden Verwaltung von Metadaten ebenfalls eine entsprechende Indexierung von diesen voraus, um eine effiziente Volltextsuche der verschiedenen Ressourcen zu ermöglichen. In Kadi4Mat kommt hierfür das System Elasticsearch [178] zum Einsatz. Bei Elasticsearch handelt es sich um eine auf Apache Lucene [179] aufbauende Suchmaschine, die eine flexible, JSON-basierte Struktur zur Speicherung von zu durchsuchenden Dokumenten verwendet und über eine HTTP-API angesprochen werden kann. Bei den Dokumenten kann es sich prinzipiell um sämtliche Art von Daten oder Metadaten handeln, die sich nach JSON konvertieren lassen. Während das System etliche Parallelen zu regulären Datenbanken aufweist, liegt der Fokus weniger auf der Persistierung der Dokumente, sondern auf deren Indexierung und der anschließenden Suche relevanter Ergebnisse unter Verwendung effizienter Suchstrategien.

Um Dokumente in Elasticsearch indexieren zu können, ist die Definition entsprechender Abbildungen (englisch: Mappings) notwendig, die den Prozess definieren, wie ein Dokument und dessen Attribute innerhalb eines Indexes gespeichert werden. Elasticsearch bietet sowohl dynamische Mappings, die automatisch aus den indexierten Dokumenten erstellt werden können (danach jedoch fix sind), als auch explizite Mappings, die im Vorfeld definiert werden müssen. Da die Metadaten der innerhalb von Kadi4Mat verwendeten Ressourcen auf einem fixen Basisschema aufbauen, kommen explizite Mappings mit passenden Datentypen zum Einsatz. Eine Besonderheit stellen jedoch die generischen Record-Metadaten dar, die sich in deren Struktur und verwendeten Datentypen stark unterscheiden können. Um die vollständige Flexibilität der Metadaten erhalten zu können, kommt ein Vorverarbeitungsschritt zum Einsatz, welcher die potenziell hierarchischen Metadaten in eine flache Struktur überführt. Diese ähnelt dem in Tabelle 5.4 dargestellten EAV-Modell, mit dem Unterschied, dass Einträge mit verschachtelten Werten, wie z. B. `person`, ignoriert werden. Lediglich der Schlüssel solcher Einträge wird als Präfix für die indexierten Metadaten simpler Datentypen verwendet, wie z. B. `person.name` als konkreter, textueller Wert. Pro Typ werden die so verarbeiteten Metadaten in separate Gruppen sortiert, um diesen ein entsprechendes, typisiertes

Mapping zuweisen zu können. Ein gekürzter Auszug des daraus resultierenden Mappings für textuelle Werte sowie dazugehöriger Schlüssel ist in Abbildung 5.7 dargestellt. Die Verwendung des `nested`-Typs stellt sicher, dass die unterschied-

```
{
    "extras_str": {
        "type": "nested",
        "properties": {
            "key": {
                "type": "text"
            },
            "value": {
                "type": "text"
            }
        }
    }
}
```

Abbildung 5.7: Auszug aus dem Mapping generischer Record-Metadaten mit textuellen Werten in Elasticsearch.

lichen Metadaten als separate Objekte betrachtet werden und damit unabhängig voneinander abgefragt werden können. Dies ermöglicht z. B. die Abfrage bestimmter Wertebereiche numerischer Metadaten in Kombination mit einer Volltextsuche textueller Metadaten innerhalb individueller Records.

Um den in PostgreSQL vorhandenen Metadatenbestand mit den Indizes in Elasticsearch synchron zu halten, werden Änderungen in ersterem mithilfe entsprechender Events auf ORM-Ebene mit den Elasticsearch-Indizes synchronisiert. Die Nachteile der dadurch entstehenden, doppelten Datenhaltung bestehen aus einem gewissen Mehraufwand, der für die Synchronisierung benötigt wird, und aus einer möglichen Asynchronität zwischen Datenbank und Suchindizes bei einer fehlerhaften Synchronisierung. Gleichzeitig entsteht hierdurch eine klare Trennung zwischen Datenhaltungs- (Datenbank) und Präsentationsschicht (Suchindizes), was auch bei der Fragestellung eine Rolle spielt, ob ein separates Suchsystem

überhaupt sinnvoll ist. PostgreSQL selbst bietet neben herkömmlichem Pattern Matching verschiedene Möglichkeiten zur Volltextsuche, z. B. unter Verwendung des in modernen PostgreSQL-Versionen bereits enthaltenen Moduls *pg_trgm* zur Unterstützung von Trigrammen (Textfragmente fixer Länge) und einer entsprechenden, unscharfen Suche (englisch: Fuzzy Search). Zusätzlich zu den bereits erwähnten Vorteilen bietet Elasticsearch als dedizierte Suchmaschine jedoch mehr Spielraum für Performanceoptimierungen [180] und die zukünftige Anpassung der Suchindizes, ohne die Struktur oder das Volumen des bestehenden Datenbestands in PostgreSQL berücksichtigen zu müssen.

5.3 Strukturierte Verwaltung von Daten

Entsprechend des in Kapitel 4 erläuterten Konzepts kommt zur Umsetzung der strukturierten Datenverwaltung ein Objektspeicher zum Einsatz, um die eigentlichen Forschungsdaten als logischer Bestandteil der Records in Kadi4Mat persistieren zu können. Bei dessen Umsetzung sind vor allem zwei Aspekte zu berücksichtigen: die Bereitstellung passender Schnittstellen zur Nutzung des Objektspeichers aus Benutzersicht sowie die Wahl des zugrunde liegenden Speichersystems, wobei erstgenannter Aspekt als Teil der in Kadi4Mat umgesetzten HTTP-API im nächsten Abschnitt näher erläutert wird. Für die eigentliche Persistierung der Daten stellt die Verwendung eines regulären, lokalen Dateisystems, das von Kadi4Mat direkt oder über ein Netzwerk zugreifbar ist, die unkomplizierteste und naheliegendste Möglichkeit dar und unterliegt keinen Einschränkungen gegenüber verschiedenen Datenformaten. Während z. B. Datenbanken prinzipiell ebenfalls zur direkten Speicherung von binären Objekten, und damit beliebigen Forschungsdaten, verwendet werden können, wird hierdurch ein regulärer und performanter Dateizugriff ermöglicht. Weiterhin wird, wie auch bereits bei Umsetzung der Suche von Metadaten, eine klare Trennung von Zuständigkeiten definiert. Um die gespeicherten Daten sowie insbesondere deren technische Metadaten, z. B. Dateigröße oder Dateityp, weiterhin effizient abfragen zu können, werden dennoch entsprechende Referenzen zu den Daten in der Datenbank hinterlegt,

die jeweils mit dem entsprechenden Datenbestand konsistent gehalten werden müssen.

Obwohl der initiale Schwerpunkt auf der Verwendung lokaler Datenspeicher liegt, können zukünftig ebenfalls andere Formen von Speichersystemen für Kadi4Mat relevant sein. Insbesondere Dienste, die auf der von Amazon S3 bereitgestellten Schnittstelle basieren, wie z. B. MinIO [181] oder Amazon S3 selbst, kommen zunehmend in Repositorien wie Zenodo oder Dataverse als primärer oder alternativer Speicher für publizierte Forschungsdaten zum Einsatz. Ein weiterer Anwendungsfall stellt die Verwaltung existierender Forschungsdaten dar, die lediglich in Form entsprechender Metadaten bzw. Verlinkungen in Kadi4Mat hinterlegt werden sollen. Diese Möglichkeit wurde bereits in Kapitel 4 u. a. im Kontext potenzieller Performanceeinschränkungen diskutiert, ist jedoch auch für Fälle relevant, in denen Kadi4Mat keine volle Kontrolle über die Daten gewährt werden kann oder soll. Dabei spielt es prinzipiell keine Rolle, ob es sich um institutionelle Speichersysteme, Cloud-Speicher oder sogar Datenbanken handelt. Diese und weitere Anwendungsgebiete erfordern zukünftig eine generische Entwicklung der Datenverwaltung, wobei entsprechende Grundvoraussetzungen bereits als Teil einer in Kadi4Mat umgesetzten Pluginschnittstelle gegeben sind, die im weiteren Verlauf dieses Kapitels näher beschrieben wird.

5.4 Benutzerschnittstellen

Wie bereits im Überblick über das Gesamtsystem erläutert, stellt Kadi4Mat sowohl eine grafische als auch programmiertechnische Schnittstelle zur Verfügung, um mit den bereitgestellten Funktionalitäten zu interagieren. Beide Schnittstellen werden in den folgenden Abschnitten vorgestellt, wobei der Schwerpunkt auf der Beschreibung der HTTP-API liegt.

5.4.1 Grafische Benutzeroberfläche

Die von Kadi4Mat bereitgestellte, webbasierte GUI stellt eine der primären Schnittstellen zur Nutzung der unterschiedlichen Funktionalitäten dar. Zur Umsetzung von dieser kommen die innerhalb von Browsern verwendeten und gängigen Webtechnologien JavaScript, HTML und CSS zum Einsatz. An etlichen Stellen wird zusätzlich das clientseitige JavaScript-Webframework Vue.js [182] verwendet. Dieses eignet sich insbesondere für die Erstellung von Single-Page-Webanwendungen (englisch: Single-Page Application, kurz SPA), kann jedoch auch für einzelne Abschnitte klassischer Webanwendungen verwendet werden, die aus mehreren, untereinander verlinkten HTML-Dokumenten bestehen. Der Vorteil des Frameworks liegt insbesondere in der klaren Trennung zwischen Daten- und Präsentationsschicht sowie in der einfachen Wiederverwendung interaktiver UI-Komponenten. Dieser Aspekt wird mit serverseitigem Rendering von HTML-Templates unter Verwendung von Jinja2 [183] kombiniert. Die webbasierte Entwicklung ermöglicht ebenfalls die Nutzung der GUI über mobile Geräte wie z. B. Smartphones oder Tablets. Dies wird mithilfe eines responsiven Designs unter Verwendung des Frameworks Bootstrap [184] realisiert. Abbildung 5.8 zeigt einen entsprechenden Screenshot, welcher die Startseite von Kadi4Mat darstellt. Einige konkretere Beispiele unterschiedlicher Ansichten der GUI werden im weiteren Verlauf der Arbeit aufgezeigt.

5.4.2 Programmierschnittstelle und automatisierte Arbeitsflüsse

Zur Unterstützung automatisierter Arbeitsflüsse, inklusive einer möglichen Laborautomatisierung, sowie programmiertechnischer Nutzung der meisten Funktionalitäten von Kadi4Mat wird eine entsprechende HTTP-API bereitgestellt. Diese kommt sowohl für interne als auch für externe Zwecke in Kadi4Mat zum Einsatz. Intern dient deren Nutzung hauptsächlich zur asynchronen Abfrage verschiedener Informationen oder Ressourcen, die anschließend innerhalb der GUI entsprechend dargestellt werden können, während die externe Nutzung dagegen sehr

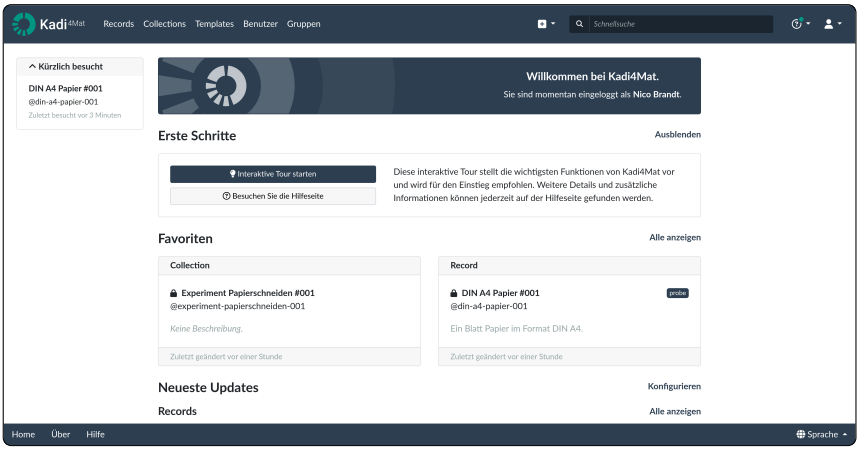

Abbildung 5.8: Screenshot der GUI von Kadi4Mat. Dargestellt ist Kadi4Mats Startseite, die unterschiedliche Möglichkeiten zur Navigation und eine Übersicht zu aktuellen Änderungen verschiedener Ressourcen bietet.

unterschiedliche Einsatzgebiete umfassen kann. Neben der Verwaltung der verschiedenen Ressourcen, Metadaten und Zugriffsrechte, wird ebenfalls die Nutzung des Objektspeichers und damit die Datenverwaltung mithilfe der HTTP-API ermöglicht. Diese orientiert sich, in stark vereinfachter Form, an der von Amazon S3 bereitgestellten Schnittstelle und erlaubt sowohl das direkte als auch das stückweise Hochladen beliebiger Daten, aktuell jedoch keine Spezifikation hierarchischer Ordnerstrukturen.

Konzeptuell basiert die HTTP-API größtenteils auf den Grundprinzipien des Representational State Transfer-Paradigmas (REST) [185], welches die Architektur verteilter, meist webbasierter Dienste mit Fokus auf der Maschine-zu-Maschine-Kommunikation beschreibt. Das Ziel des Paradigmas liegt darin, die bereits vorhandene Infrastruktur des World Wide Webs und Semantik von HTTP auszunutzen, was den positiven Nebeneffekt hat, dass auch existierende Dienste oft ohne Anpassungen bereits größtenteils REST-konform sind. Das Hauptmerkmal von REST liegt in der Forderung nach einer einheitlichen Schnittstelle. Die von einem Dienst angebotenen Ressourcen müssen dazu über eindeutige und möglichst konsistente Identifikatoren abrufbar sein, wobei in webbasierten Diensten URLs dafür

zum Einsatz kommen, die häufig auch als Endpunkte bezeichnet werden. Die Repräsentation der Ressourcen kann in verschiedenen Formaten erfolgen, wobei überwiegend, so wie auch in Kadi4Mat, JSON als Austauschformat zum Einsatz kommt. Weiterhin müssen die von einem Client empfangenen Nachrichten selbstbeschreibend sein, was nicht nur die beschriebenen Ressourcen selbst betrifft, sondern auch die Verwendung standardisierter Methoden zur Manipulation der Ressourcen. Bei Verwendung von HTTP bieten sich dessen vorhandene Methoden für diesen Zweck an, wie z. B. die Methode `POST` zur Erstellung neuer Ressourcen. Zuletzt spielt das HATEOAS-Prinzip (englisch: Hypermedia as the Engine of Application State) eine wichtige Rolle. Dieses setzt voraus, dass ausgehend von einer initialen Anfrage alle weiteren, für eine bestimmte Ressource relevanten Endpunkte als Teil der Nachrichten zurückgeliefert werden. Während das REST-Paradigma durch eine geringe Komplexität und breite Unterstützung hervorsticht, wird im Vergleich zu verwandten Konzepten zur Umsetzung von webbasierten APIs, wie z. B. GraphQL [186] oder gRPC [187], üblicherweise eine geringere Effizienz geboten. Letztgenanntes Beispiel bietet sich u. a. auch für Aspekte wie Echtzeitanwendungen oder bidirektionale Kommunikation an und kann daher als alternative Implementierung zukünftig ebenfalls relevant sein.

Zur Umsetzung von REST ist jede Ressource in Kadi4Mat eindeutig über eine entsprechende URL adressierbar, die neben dem Ressourcentyp jeweils den innerhalb einer Instanz von Kadi4Mat automatisch generierten und eindeutigen Identifikator (ID) enthält. Beispielsweise kann der Endpunkt `/api/records/1` (bei ausschließlicher Betrachtung des Pfades der entsprechenden URL) dazu verwendet werden, die Metadaten des Records mit ID 1 in einer JSON-basierten Repräsentation abzufragen. Abbildung 5.9 zeigt einen beispielhaften und stark gekürzten Auszug der zurückgelieferten Metadaten bei Verwendung dieses Endpunkts mithilfe der `GET`-Anfragemethode. Neben den eigentlichen Metadaten des Records, welche in diesem Beispiel lediglich dessen ID und Identifier umfassen, werden zur Umsetzung von HATEOAS zusätzliche Links (`_links`) in Form von URLs bereitgestellt, die z. B. zum Abrufen weiterer Informationen wie den zugehörigen Dateien des Records verwendet werden können. Durch die einfach verständliche Semantik der verschiedenen HTTP-Methoden und Endpunkte sowie

```
{
    "id": 1,
    "identifier": "record-1",
    "_links": {
        "files": "https://k4m.example/api/records/1/files"
    }
}
```

Abbildung 5.9: Auszug aus der JSON-basierten Repräsentation eines Records unter Verwendung der HTTP-API von Kadi4Mat.

der Umsetzung von HATEOAS, lassen sich REST-APIs häufig auch ohne umfangreiche Dokumentation und unter Verwendung verschiedener Clients navigieren, was einer der größten Vorteile der Technologie darstellt.

Während bei interner Nutzung die Authentifizierung bzw. Autorisierung zur Nutzung der HTTP-API über bereits vorhandene Cookies stattfindet, die eingeloggte Benutzer über mehrere Anfragen hinweg authentifizieren, werden für die externe Nutzung aktuell zwei verschiedene Mechanismen unterstützt: die Verwendung von personenbezogenen Zugriffstokens (englisch: Personal Access Token, kurz PAT) oder von auf OAuth 2.0 basierenden Schnittstellen zur (semi-)automatischen Abfrage entsprechender Tokens. Beide Arten von Tokens lassen sich auf dieselbe Weise mithilfe eines passenden `Authorization`-HTTP-Headers verwenden und sind immer genau einem Benutzer und seinen jeweiligen Zugriffsrechten zugeordnet. PATs lassen sich über die GUI von Kadi4Mat verwalten und erlauben die direkte Verwendung der HTTP-API innerhalb unterschiedlicher Programmiersprachen und Clients, wobei die Gültigkeit und der Geltungsbereich (englisch: Scope) der Tokens benutzerdefiniert ist. Die Verwendung von OAuth 2.0 erlaubt dagegen insbesondere die Integration von Kadi4Mat innerhalb externer, webbasierter Anwendungen. Abbildung 5.10 zeigt einen schematischen Ablauf zwischen einer solchen Anwendung und Kadi4Mat unter Verwendung der entsprechenden Schnittstellen. Bei diesem handelt es sich um den sogenannten Authorization Code Grant, welcher die gängigste und aktuell in Kadi4Mat implementierte Variante des OAuth 2.0-Protokolls darstellt. Hierbei wird, nach einmaliger Autorisierung der externen Anwendung in Kadi4Mat durch einen entsprechenden Benutzer,

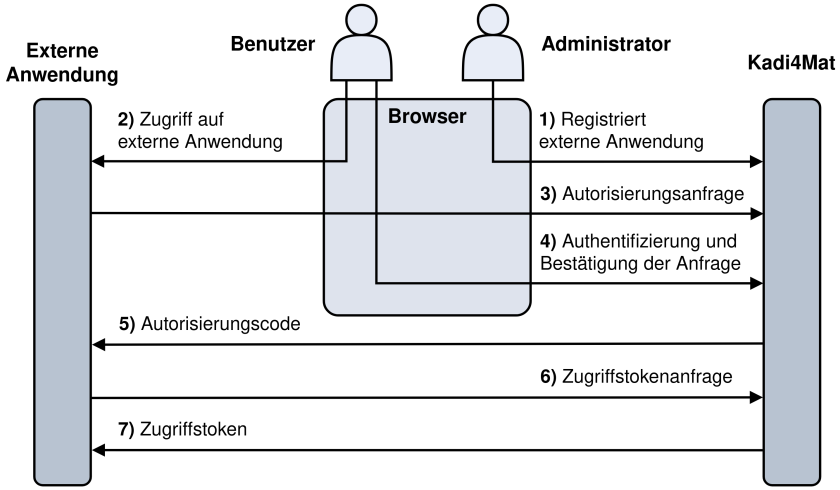

Abbildung 5.10: Schematischer Ablauf eines typischen OAuth 2.0-Flows unter Verwendung des Authorization Code Grants. Ein Administrator der externen Anwendung registriert diese in einer konkreten Kadi4Mat-Instanz, sodass eine entsprechende Autorisierungsanfrage ermöglicht wird. Wird diese vom Benutzer bestätigt, was ein Konto in beiden Anwendungen voraussetzt, kann unter Verwendung eines Autorisierungscodes letztendlich ein Zugriffstoken generiert werden, welches die externe Anwendung für zukünftige, autorisierte Anfragen verwenden kann.

ein Autorisierungscode generiert, der in einem weiteren Schritt durch das eigentliche Zugriffstoken ausgetauscht wird. Letzteres bleibt dem Benutzer der externen Anwendung üblicherweise verborgen. Im Gegensatz zu der bereits in Kapitel 4 erläuterten Nutzung von OAuth 2.0 im Rahmen von OIDC zielt dieser Anwendungsfall primär auf die Autorisierung ab. Der Vorteil bei der Nutzung von OAuth 2.0 besteht darin, dass es sich bei der externen Anwendung prinzipiell ebenfalls um eine weitere Instanz von Kadi4Mat handelt kann. Hierdurch können zukünftig Funktionalitäten wie instanzübergreifende Suchen oder der einfache Austausch von Ressourcen ermöglicht werden, sofern ein entsprechender Benutzer ein Konto in beiden Kadi4Mat-Instanzen besitzt.

Um die Nutzung und insbesondere Integration der HTTP-API in existierende Arbeitsflüsse unterstützen zu können, wurde zusätzlich die Programmbibliothek kadi-apy [188] entwickelt. Diese ist mithilfe der Programmiersprache Python

umgesetzt und bietet eine systemunabhängige und objektorientierte Arbeitsweise zum Erstellen, Abrufen und Modifizieren der durch Kadi4Mat als Teil der HTTP-API bereitgestellten Ressourcen. Die zur Interaktion mit einer konkreten Instanz von Kadi4Mat notwendigen Einstellungen, wie deren URL und ein zur Autorisierung benötigtes PAT, lassen sich optional über eine zentrale Konfigurationsdatei festlegen. Neben der Verwendung der Bibliothek innerhalb von Python-Code wird ebenfalls eine Kommandozeilenschnittstelle (englisch: Command-Line Interface, kurz CLI) bereitgestellt. Während diese im Vergleich weniger Flexibilität bietet als die direkte Nutzung in Python, wird hierdurch eine einfache Integration mit anderen Programmier- oder Skriptsprachen ermöglicht.

5.5 Integration existierender Systeme und benutzerdefinierte Arbeitsabläufe

Um die Integration existierender Systeme in Kadi4Mat ermöglichen zu können, beinhaltet das System eine Pluginschnittstelle. Diese ermöglicht die Umsetzung von Modifikationen oder die Erweiterung existierender Funktionalitäten, ohne dazu vorhandenen Code im Kern anpassen zu müssen oder konkrete Implementierungsdetails zu kennen. Die Schnittstelle baut auf der Bibliothek pluggy [189] auf, die einen einfachen Mechanismus zur Registrierung und Ausführung von Plugins in Python bereitstellt und bereits in etlichen Produktivsystemen zum Einsatz kommt. Konzeptuell setzt pluggy auf eine lose Kopplung zwischen Anwendung und Plugins, indem die zu erweiternde Anwendung lediglich Spezifikationen, sogenannte Plugin-Hooks, in Form einfacher Funktionsdeklarationen vorgibt. Plugins können anschließend eine beliebige Anzahl solcher Hooks implementieren. Durch den Aufruf der Hooks an vordefinierten Stellen in der Anwendung, z. B. bei deren Initialisierung, bei der grafischen Darstellung einer Seite oder bei Bearbeitung einer bestimmten Anfrage, können die Implementierungen bestimmter Hooks nacheinander aufgerufen werden. Während damit primär die Verwendung von in externen Python-Paketen implementierten Funktionalitäten ermöglicht wird, lassen sich sämtliche Hooks auch zur Umsetzung von Kernfunktionalitäten in

Kadi4Mat nutzen. Dies bietet den Vorteil, dass die entsprechenden Schnittstellen einheitlich und konsistent verwendet werden können.

Wie bereits in Kapitel 4 erläutert, bietet sich insbesondere für die Publizierung von Forschungsdaten die Integration existierender Systeme bzw. Repositorien an. Um generell eine direkte Publizierung von Records und Collections aus Kadi4Mat heraus ermöglichen zu können, unabhängig vom konkreten Repositorium, wird von der erläuterten Pluginschnittstelle Gebrauch gemacht. Hierzu werden unterschiedliche Hooks bereitgestellt, die zur Registrierung und Konfiguration entsprechender Repositorien sowie zur eigentlichen Publizierung der Forschungsdaten zum Einsatz kommen. Der letztgenannte Aspekt wird dabei innerhalb eines Hintergrundprozesses unter Verwendung von Celery gekapselt, um insbesondere die Publizierung größerer Datenvolumen zu unterstützen. Die einzige Voraussetzung zur Nutzung eines Repositoriums ist aktuell die Verfügbarkeit einer HTTP-API unter Verwendung des OAuth 2.0-Protokolls, wie auch von Kadi4Mat selbst umgesetzt, wobei in diesem Fall die in Abbildung 5.10 dargestellten Rollen von Kadi4Mat und der externen Anwendung vertauscht sind. Dies ermöglicht die Integration des entsprechenden Repositoriums ohne die Notwendigkeit manueller Verwaltung von Zugriffstokens aus Benutzersicht.

Als konkreter Anwendungsfall wurde eine Anbindung des Systems Zenodo umgesetzt, da dieses als öffentlich zugängliches und interdisziplinäres Repositorium einen idealen Ausgangspunkt für die Publizierung heterogener Forschungsdaten bietet und ebenfalls die Vergabe von PIDs in Form von DOIs unterstützt. Das entsprechende Plugin ist bereits fester Bestandteil von Kadi4Mat und muss für dessen Verwendung lediglich von einem Systemadministrator konfiguriert werden. Da innerhalb von Zenodo das DataCite Metadata Schema zum Einsatz kommt, ist ein entsprechender Crosswalk zur Abbildung von Metadaten notwendig, der ähnlich zu dem in Tabelle 5.2 dargestellten Beispiel für unterstützte Elemente des Basisschemas (z. B. Titel, Beschreibung oder Lizenz) in Kadi4Mat umgesetzt ist. Für die Bündelung der zu publizierenden Forschungsdaten kommt das RO-Crate-Format zum Einsatz. Neben den eigentlichen Forschungsdaten, die je nach publiziertem Ressourcentyp aus einem oder mehreren Records stammen können,

sind innerhalb des entsprechenden Archivs zusätzlich die vollständigen Metadaten aller zu publizierenden Ressourcen in unterschiedlichen Exportformaten enthalten, die im späteren Verlauf dieses Kapitels näher beschrieben werden. Zwar bieten RO-Crates ebenfalls Möglichkeiten zur direkten Spezifikation von Metadaten als Teil der im Archiv enthaltenen JSON-LD-Metadatendatei, diese beschränkt sich jedoch hauptsächlich auf das Schema.org-Vokabular und eignet sich, neben der Beschreibung der Archivstruktur, lediglich für grundlegende Metadaten des Basisschemas von Kadi4Mat.

Ein weiteres Beispiel zur Nutzung der Pluginschnittstelle stellt die Integration von Terminologieservices in Kadi4Mat dar. Die bereits vorhandene Funktionalität zur Spezifikation von Termen als Teil individueller, generischer Metadaten wurde dazu um eine generische Suchfunktion für entsprechende Terme erweitert. Diese lässt sich über die GUI von Kadi4Mat verwenden und leitet die eigentliche Suchanfrage an einen externen Dienst weiter. Wird ein passender Term gefunden und vom Benutzer ausgewählt, kann dessen IRI direkt in die generischen Metadaten übernommen werden. Ein entsprechendes Plugin wurde für den bereits vorgestellten Terminologieservice der TIB umgesetzt und kann in gleicher Weise wie das Plugin für Zenodo aktiviert bzw. konfiguriert werden.

Ähnlich können auch weitere Anwendungsfälle unter Verwendung der Pluginschnittstelle sowie passender APIs umgesetzt werden, wie z. B. die bereits erläuterte, generische Integration unterschiedlicher Speichersysteme. Neben der Nutzung existierender Systeme wird durch die bereitgestellte Pluginschnittstelle ebenfalls die in Kapitel 4 erläuterte Notwendigkeit zur Umsetzung benutzerdefinierter Arbeitsabläufe ermöglicht. Hierzu werden unterschiedliche Plugin-Hooks bereitgestellt, die u. a. für benutzerdefinierte Datenvisualisierungen, zur Reaktion auf Änderungen von Ressourcen oder zur Personalisierung einer konkreten Instanz von Kadi4Mat eingesetzt werden können. Letzterer Aspekt wird in einfacher Form ebenfalls anhand verschiedener Konfigurationsmöglichkeiten auf benutzer- oder instanzweiter Ebene unterstützt.

5.6 Authentifizierungs- und Autorisierungsinfrastruktur

Da die Wahl geeigneter Authentifizierungsmöglichkeiten stark von der jeweiligen Umgebung abhängig ist, innerhalb der eine Instanz von Kadi4Mat installiert und verwendet werden soll, werden verschiedene Mechanismen zur Verfügung gestellt, die sich zentral durch einen zuständigen Systemadministrator konfigurieren lassen. Die einfachste Art der Authentifizierung besteht in der Verwendung lokaler Benutzerkonten, deren Zugangsdaten unter Berücksichtigung entsprechender Sicherheitsstandards komplett von Kadi4Mat verwaltet werden. Der Vorteil lokaler Benutzerkonten besteht in ihrer einfachen Handhabung, da keine externen Systeme sowie entsprechende Konfigurationen notwendig sind. Gleichzeitig ist die Verwendung existierender Konten bzw. SSO oft wünschenswert, um eine zentrale und einfach zu verwaltende Authentifizierung gewährleisten zu können, sowie häufigen Sicherheitsproblemen wie der Wiederverwendung von Passwörtern entgegenzuwirken. Hierzu lassen sich die bereits in Kapitel 4 erläuterten Mechanismen LDAP, in unterschiedlichen Implementierungen, sowie Shibboleth zur Authentifizierung integrieren. Die erste Option ist geeignet, wenn bereits eine existierende LDAP-Installation im Einsatz ist, während sich Shibboleth insbesondere durch die Möglichkeit zur Föderation unterschiedlicher Identity Provider auszeichnet. Auch wenn im aktuellen Stand noch nicht in Kadi4Mat umgesetzt, kann ebenfalls die zukünftige Integration von OIDC relevant sein, um Benutzerkonten existierender Dienste wie z. B. ORCID zur Authentifizierung nutzen zu können.

Zur Verwaltung von Zugriffsrechten auf verschiedene Ressourcen existiert eine speziell auf Kadi4Mat zugeschnittene Autorisierungsinfrastruktur. Diese basiert hauptsächlich auf RBAC, wobei pro Ressourcentyp unterschiedliche mögliche Rollen definiert sind. Jeder Rolle sind wiederum eine oder mehrere Aktionen zugeordnet, beispielsweise zum Lesen oder Bearbeiten einer individuellen Ressource. Wie zu Beginn dieses Kapitels erwähnt, ist die Vergabe von Rollen sowohl an einzelne Benutzer als auch an Gruppen mehrerer Benutzer möglich. Konzeptuell

folgt die Überprüfung von Zugriffsrechten der Funktionsweise von ACLs, sodass z. B. einzelnen Mitgliedern einer Gruppe andere Zugriffsrechte gewährt werden können als der Gruppe selbst. Um in speziellen Fällen ebenfalls unabhängig von Rollen bestimmte Zugriffsrechte gewähren zu können, kommt ebenfalls ABAC in einfacher Version zum Einsatz, z. B. um nur dem Ersteller einer Ressource deren endgültige Löschung zu gestatten. In diesem Kontext kann zukünftig ebenfalls die Verwendung von aus externen Systemen stammenden Attributen in Betracht gezogen werden, um z. B. abhängig von durch Shibboleth übertragenen Instituts-zugehörigkeiten automatisch bestimmte Zugriffsrechte vergeben zu können.

5.7 Beispielhafter Anwendungsfall

Um die beschriebenen Ressourcen und Funktionalitäten von Kadi4Mat detaillierter und vor allem praxisorientierter erläutern zu können, wird in diesem Abschnitt ein beispielhafter Anwendungsfall vorgestellt. Bei diesem handelt es sich um ein simples und fiktives Experiment, dessen Durchführung in Abbildung 5.11 dargestellt ist. Ausgangspunkt von diesem bildet eine Schere als beispielhaftes

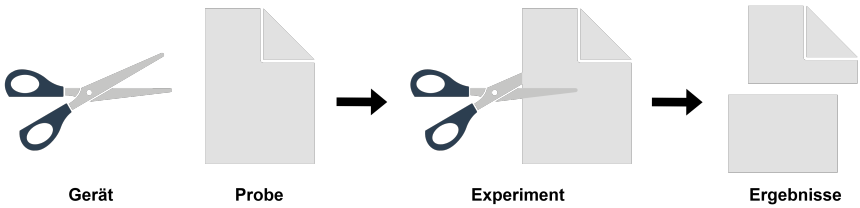

Abbildung 5.11: Beispielhafte Durchführung eines simplen Experiments unter Verwendung eines Geräts und einer Probe.

Gerät, sowie ein Blatt Papier als beispielhafte Probe, wobei letztere mithilfe des Geräts bearbeitet werden soll. Die Durchführung des Experiments besteht im Schneiden des Blatt Papiers, welches die ursprüngliche Probe in einen neuen Zustand in Form zweier Hälften des Papiers überführt. Dieser Anwendungsfall stellt einen sehr vereinfachten Ablauf eines Experiments unter Verwendung physischer

Versuchsobjekte dar, lässt sich jedoch größtenteils ebenfalls auf digitale Forschungsprozesse übertragen, wie z. B. die Analyse existierender Forschungsdaten unter Verwendung einer Analysesoftware und entsprechender Eingabeparameter. Das Ziel bei der Durchführung des Experiments besteht darin, von Anfang an alle beteiligten Objekte und Prozesse innerhalb von Kadi4Mat aufzuzeichnen, wobei angenommen wird, dass keine bereits erstellten und passenden Ressourcen in der verwendeten Instanz von Kadi4Mat existieren. Zur Verwendung von Kadi4Mats Funktionalitäten wird in diesem Beispiel hauptsächlich von der GUI Gebrauch gemacht.

5.7.1 Vorbereitung des Experiments

Zur Vorbereitung des Experiments werden zuerst alle Objekte, welche die Ausgangsbasis des Experiments darstellen, in Form von Records in Kadi4Mat hinterlegt. Hierbei handelt es sich um das Gerät und die Probe, die jeweils ein physisches Objekt repräsentieren und daher lediglich in einer digitalisierten Form repräsentiert werden können. Die entsprechenden Records bestehen daher nur aus Metadaten und bilden somit digitale Zwillinge der eigentlichen Objekte. Bei der Erstellung eines neuen Records über die GUI von Kadi4Mat werden zuerst dessen Metadaten spezifiziert. Im Falle von Records wird hierbei zwischen den grundlegenden und den generischen Metadaten unterschieden. Die Angabe von ersteren sind in Abbildung 5.12 dargestellt, wobei die Probe des Experiments als konkretes Beispiel verwendet wird. Bei der Angabe des Titels der Probe kommt eine nutzerlesbare Bezeichnung gefolgt von einer fiktiven Laufnummer zum Einsatz, die ebenfalls für den entsprechenden Identifier verwendet wird. Dieser lässt sich optional automatisch anhand des spezifizierten Titels generieren. Für die Freitextbeschreibung wird ein Editor zur Verfügung gestellt, der unterschiedliche Funktionen zur einfachen Spezifikation von Markdown-Syntax sowie eine entsprechende Vorschau bietet. Metadaten wie der Typ oder die Tags eines Records lassen sich entweder direkt spezifizieren oder anhand vorhandener Metadaten anderer Records suchen und auswählen, während die möglichen Werte für Lizenzen aus einer vordefinierten Liste stammen. Mithilfe dieser Funktionalitäten wird

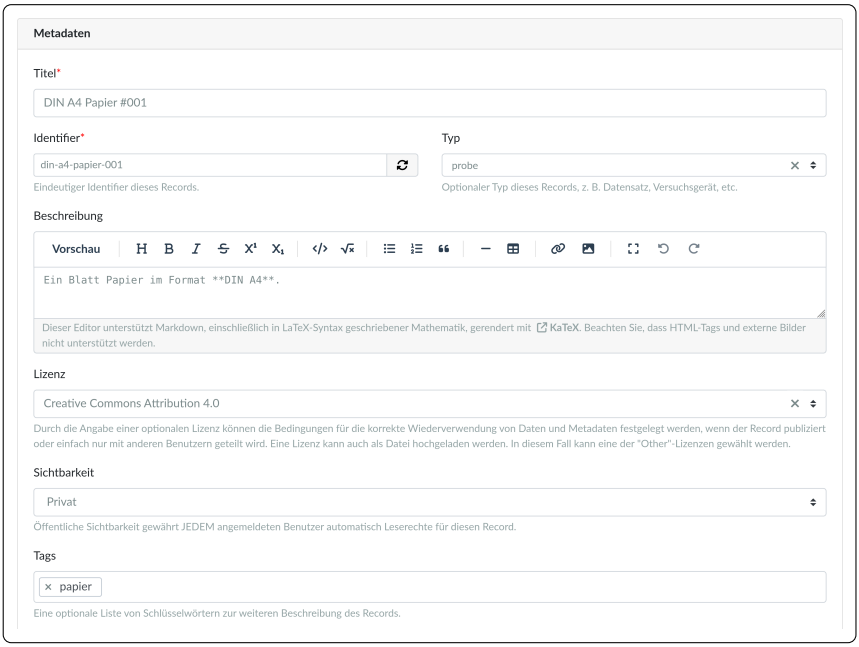

Abbildung 5.12: Angabe grundlegender Metadaten bei der Erstellung eines Records über die GUI von Kadi4Mat.

auch ohne Verwendung eines Templates bereits die manuelle Spezifikation von Metadaten vereinfacht.

Die Angabe generischer Record-Metadaten erfolgt über einen separaten Editor, der in Abbildung 5.13 dargestellt ist. Dieser erlaubt die Angabe entsprechender Schlüssel-Wert-Paare unterschiedlicher Typen, sowie verschiedene Funktionalitäten zum Manipulieren, Kopieren oder Visualisieren der Metadatenelemente und -struktur. Pro Typ werden entsprechende Eingabefelder zur Verfügung gestellt, wie z. B. das in Abbildung 5.13 gezeigte, reguläre Eingabefeld für textuelle Werte (`Format`) oder die Kombination zweier Eingabefelder zur Angabe eines numerischen Werts und einer optionalen Einheit (`Grammgewicht`). Ebenfalls dargestellt ist die Angabe eines geschachtelten Metadatums (`Maße`), das verwendet wird, um die zwei folgenden, numerischen Metadaten (`Breite` und `Höhe`) unter einem

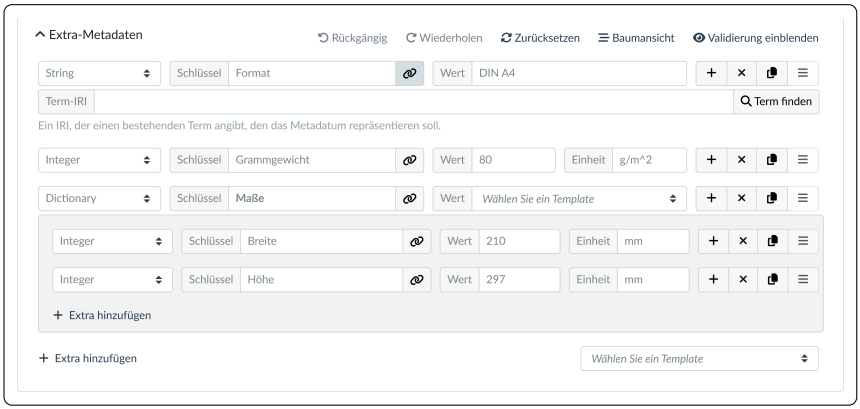

Abbildung 5.13: Angabe generischer Metadaten bei der Erstellung eines Records über die GUI von Kadi4Mat.

einzigen Schlüssel zu gruppieren. Das eigentliche Eingabefeld für Werte verschachtelter Metadaten kann ebenfalls dazu verwendet werden, um in Templates abgelegte, generische Metadaten laden zu können, was auch auf oberster Ebene der Metadatenhierarchie möglich ist. Weiterhin unterstützt der Editor die Angabe der bereits erläuterten Terme bzw. Konzepte in Form passender IRIs, wobei die entsprechende Option für das erste in Abbildung 5.13 dargestellte Metadatenelement aufgezeigt ist, inklusive der Möglichkeit zur Suche bereits bestehender Terme in einem Terminologieservice. Ähnlich ist die Angabe optionaler Validierungsanweisungen möglich, was vor allem bei der Spezifikation von Templates unter Verwendung desselben Editors zum Einsatz kommt. In diesem Beispiel werden die generischen Metadaten jedoch ohne Verwendung von Templates oder unter Berücksichtigung vorhandener Metadatenschemata spezifiziert, wobei letzterer Aspekt insbesondere in den Ingenieurwissenschaften aufgrund fehlender Standards einen häufigen Anwendungsfall darstellt.

Nach Erstellung des Records lassen sich optional die zu den Metadaten zugehörigen Daten hochladen, was jedoch für die beiden hier relevanten Records, die lediglich aus Metadaten bestehen, nicht erfolgt. Eine Übersicht über den resultierenden Record ist in Abbildung 5.14 dargestellt. Neben den bei der Erstellung

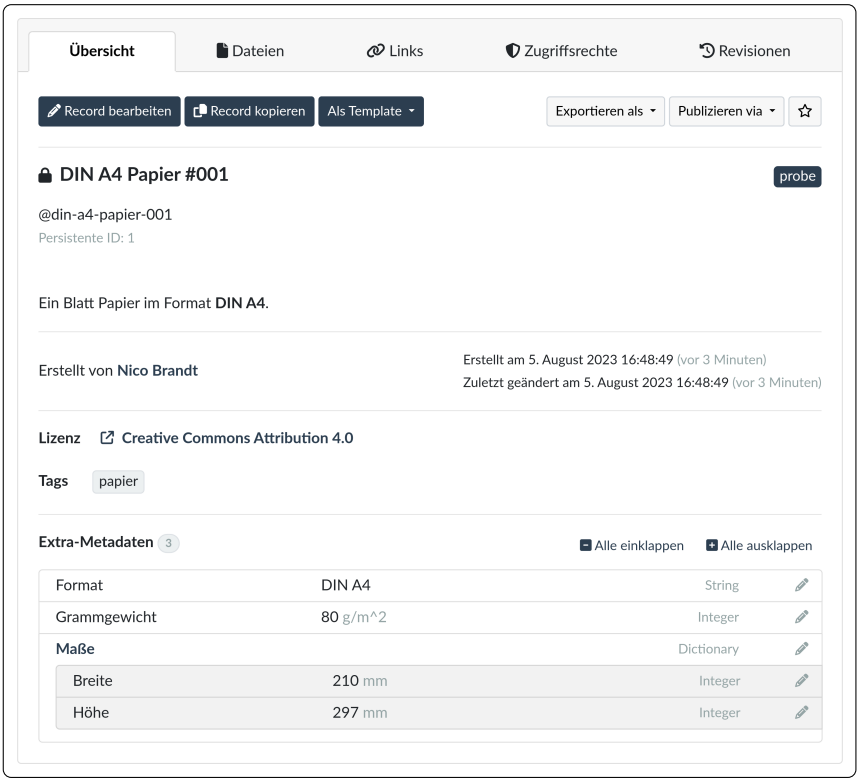

Abbildung 5.14: Screenshot einer Übersicht eines Records in Kadi4Mat.

spezifizierten Metadaten sind ebenfalls automatisch hinzugefügte Metadaten zu erkennen, wie z. B. der Ersteller des Records oder unterschiedliche Zeitstempel. Die generischen Metadaten des Records sind gesondert im unteren Teil der Übersicht in einer hierarchischen Ansicht dargestellt. Die für diesen Record spezifizierten Metadaten enthalten das Format des Papiers als textueller Wert, der eine spezifische DIN-Norm referenziert, das Grammgewicht als numerischer Wert mit zusätzlicher Einheit, sowie die Maße des Papiers als Kombination aus den numerischen Werten für Breite und Höhe, auch wenn deren Angabe technisch

gesehen redundant ist. Durch die Navigationsleiste, die im oberen Teil der in Abbildung 5.14 dargestellten Übersicht abgebildet ist, lassen sich weitere Ansichten des Records einsehen. Diese umfassen die Dateien des Records, Verlinkungen mit Collections oder anderen Records, Zugriffsrechte des Records und Revisionen der Metadaten. Weiterhin erlaubt die Übersichtsseite unterschiedliche Interaktionen mit dem aktuellen Record, was neben dessen Bearbeitung ebenfalls Export- und Publizierungsfunktionalitäten beinhaltet, die im weiteren Verlauf des Beispielexperiments relevant sind.

5.7.2 Durchführung des Experiments

Während die eigentliche Durchführung des Experiments, das Schneiden des Papiers, anhand der entsprechenden physischen Objekte manuell erfolgt, kann für das Experiment selbst ebenfalls ein Record in Kadi4Mat angelegt werden, welcher den entsprechenden Prozess beschreibt und dessen Parameter und Versuchsbedingungen enthält. Dies kann analog zu den davor erstellten Records erfolgen, welche die Probe und das Gerät repräsentieren. Auch dieser Record beinhaltet lediglich Metadaten, jedoch besteht neben dem Hochladen existierender Daten zusätzlich die Möglichkeit, skizzenhafte Beschreibungen direkt über die GUI von Kadi4Mat erstellen und als Bilddatei hochladen zu können. Dadurch lassen sich z. B. Versuchsaufbauten grob dokumentieren, was insbesondere für solche Arten von Records sinnvoll sein kann und mit der Verwendung mobiler Geräte zur Nutzung von Kadi4Mat in einem Labor kombiniert werden kann.

Zwar entstehen bei diesem Beispiel keine Messdaten, die ebenfalls als separate Records hinterlegt werden könnten, jedoch werden zwei Ergebnisse bzw. neue Proben erzeugt, in Form zweier Hälften des geschnittenen Papiers. Diese lassen sich ähnlich zur ursprünglichen Probe als Metadaten-Records repräsentieren, womit für sämtliche Objekte und Prozesse des Experiments ein digitales Gegenstück existiert. Um diese geeignet in Beziehung setzen zu können, bietet sich die Verwendung von Record-Links an, die während der Durchführung oder am Ende des Experiments über die GUI von Kadi4Mat erstellt werden können. Der daraus

resultierende Graph ist in Abbildung 5.15 dargestellt. Ausgehend vom Experiment-

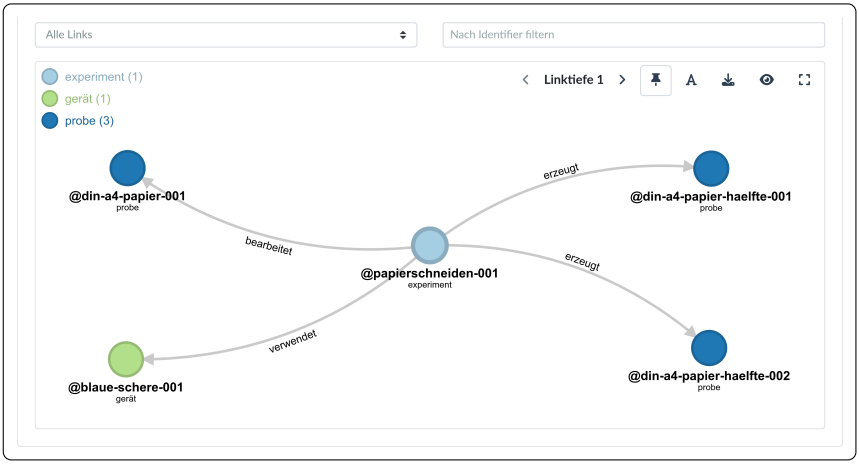

Abbildung 5.15: Interaktive Visualisierung der im Beispielexperiment involvierten Records sowie deren Verlinkungen untereinander. Die unterschiedlichen Record-Typen sind durch entsprechende Farben gekennzeichnet.

Record, welcher im Zentrum des Graphen als hellblauer Knoten hervorgehoben ist, existiert pro Record ein ausgehender Link. Jeder Link ist durch dessen Namen beschrieben, welcher die Semantik der jeweiligen Beziehung beschreibt und, wie bereits erwähnt, optional durch einen Term in Form eines entsprechenden IRIs näher spezifiziert werden kann. Diese Art von Graph lässt sich direkt innerhalb der GUI von Kadi4Mat generieren und erlaubt eine interaktive Visualisierung aller Links eines bestimmten Records bis zu einer bestimmten, maximalen Linktiefe.

Um unabhängig von etwaigen Verlinkungen der Records untereinander das Experiment als Gesamtheit gruppieren zu können, eignet sich zudem die Verwendung einer entsprechenden Collection. Die Erstellung von dieser mithilfe der GUI von Kadi4Mat erfolgt ähnlich wie die in Abbildung 5.12 dargestellte Erstellung eines Records, mit Ausnahme einiger Metadaten, da Collections primär der Organisation dienen. Anschließend lassen sich sämtliche Records mit der Collection verlinken, was auch bereits bei Erstellung der einzelnen Records möglich ist,

vorausgesetzt die entsprechende Collection wurde bereits im Vorfeld für das Experiment erstellt. Neben der Organisation der experimentellen Objekte und Prozesse, sind Collections auch bei der Suche vorhandener Records nützlich. Ein Beispiel der entsprechenden Suchmaske für Records innerhalb der GUI von Kadi4Mat ist in Abbildung 5.16 dargestellt. Diese ermöglicht die Angabe von Suchanfragen

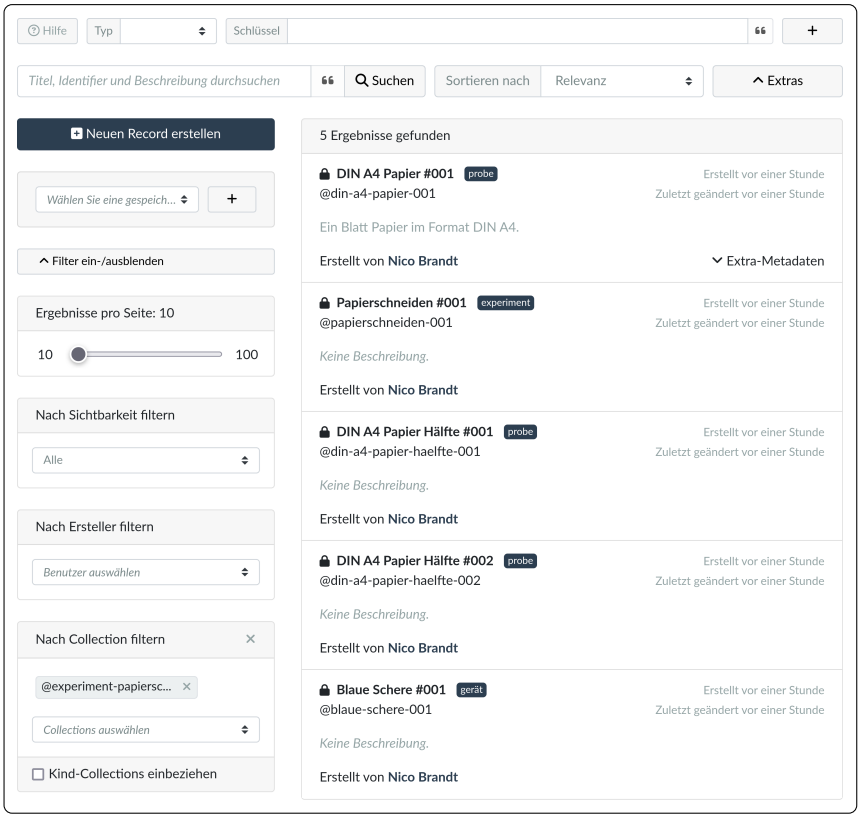

Abbildung 5.16: Interaktive Suchmaske für Records in Kadi4Mat mit unterschiedlichen Filtermöglichkeiten.

innerhalb der grundlegenden Metadaten von Records sowie das Filtern unterschiedlicher Attribute. Letztere Funktionalität erlaubt u. a. die Auswahl der zuvor

erstellten Collection, sodass unabhängig von den Inhalten der einzelnen Records in Kombination mit weiteren Anfragen oder Filtern deren gezielte Suche ermöglicht wird, was insbesondere bei komplexeren Experimenten relevant sein kann. Die zusätzliche Suchmaske, die im oberen Teil von Abbildung 5.16 dargestellt ist, ermöglicht außerdem Anfragen unter Verwendung verschiedener Typen, Schlüssel und Wertebereiche der generischen Metadaten von Records.

Einen weiteren, relevanten Aspekt bei der digitalen Erfassung des Experiments können die Zugriffsrechte auf die verschiedenen Ressourcen darstellen, da die in diesem Beispiel erstellten Records bisher nur von ihrem Ersteller einsehbar und verwaltbar sind. Unter Annahme einer fiktiven Arbeitsgruppe, die für das Experiment zuständig ist oder auf dessen digitalisierte Form Zugriff haben soll, lässt sich analog eine Gruppe mit entsprechenden Mitgliedern innerhalb von Kadi4Mat erstellen. Dieser kann Zugriff auf sämtliche Ressourcen gewährt werden, was ebenfalls direkt bei Erstellung der einzelnen Ressourcen möglich ist, vorausgesetzt die Gruppe existiert zu diesem Zeitpunkt bereits. Collections bieten zudem Funktionalitäten, um ebenfalls im Nachhinein die einfache Verwaltung von Zugriffsrechten aller verlinkter Records über die GUI von Kadi4Mat zu ermöglichen.

5.7.3 Nachnutzung und Interoperabilität

Die digitale Erfassung des gesamten Experiments ermöglicht nicht nur eine strukturierte Dokumentation sämtlicher Arbeitsschritte, sondern auch die Nachnutzung einzelner Records. Beispielsweise könnte eine weitere Probe in einem ähnlich aufgebauten Folgeexperiment mit demselben Gerät bearbeitet werden. Während der entsprechende Geräte-Record unverändert wiederverwendet werden kann, müssen für Probe, Experiment sowie gegebenenfalls dadurch entstehende Ergebnisse neue Records erzeugt werden, sofern die Struktur des vorherigen Experiments beibehalten werden soll. Für diese Zwecke bietet sich die Verwendung von Record-Templates an, um insbesondere über die GUI von Kadi4Mat manuell durchgeführte, sich wiederholende Schritte vereinfachen zu können. Neben vorgegebenen Metadaten oder entsprechenden Platzhaltern, inklusive generischer

Metadaten, lassen sich ebenfalls vordefinierte Zugriffsrechte sowie verlinkte Records oder Collections in dieser Art von Template spezifizieren. Bei einer größeren Anzahl ähnlich durchgeführter und komplexerer Experimente kann als weiterer Schritt zusätzlich die Nutzung der HTTP-API von maßgeblichem Vorteil sein. Auch wenn eine manuelle Durchführung des Experiments notwendig ist, so wie in diesem Beispiel, ist zumindest eine Teilautomatisierung unter Vorbereitung entsprechender Records für statische Geräte oder Proben denkbar.

Die externe Nachnutzung des Experiments wird insbesondere durch die in Kadi4Mat enthaltenen Export- und Publizierungsfunktionalitäten für Records und Collections ermöglicht. Für beide Ressourcentypen wird aktuell der Export in Form von QR-Codes, JSON, RDF und RO-Crates unterstützt, wobei Records zusätzlich den Export von PDF-basierten Dokumenten als nutzerlesbare Dokumentation unabhängig von Kadi4Mats GUI erlauben. QR-Codes verlinken auf die entsprechende Übersichtsseite einer Ressource und können z. B. dazu verwendet werden, physische Objekte und digitale Gegenstücke in Form von Records miteinander zu verknüpfen. Der Export von Ressourcen in JSON oder RDF ermöglicht dagegen deren maschinenlesbare Nutzung. Ersteres Format unterliegt dabei keiner standardisierten Struktur, sondern orientiert sich an der bereits für die HTTP-API von Kadi4Mat verwendeten, JSON-basierten Serialisierung der unterschiedlichen Metadaten. Mithilfe von RDF kann unter Verwendung entsprechender Vokabulare zusätzlich die semantische Interoperabilität der Metadaten ermöglicht werden, wobei zur Serialisierung der Tripel das gängige Turtle-Format verwendet wird. Zur Definition der Tripel kommen hauptsächlich die von Schema.org und RDFS bereitgestellten Vokabulare zum Einsatz, um die grundlegenden Metadaten und möglichen Verlinkungen der Ressourcen spezifizieren zu können. Abbildung 5.17 zeigt ein vereinfachtes Beispiel des RDF-Exports eines fiktiven Records unter Verwendung der in Abbildung 5.5 dargestellten, generischen Metadaten, welche durch das Prädikat `rdfs:isDefinedBy` als Teil des Records definiert werden. Auch wenn für die Records des Beispielexperiments nicht anwendbar, kann u. a. bei den generischen Metadaten von den optional spezifizierbaren Termen in Form entsprechender IRIs Gebrauch gemacht werden. Für generische Metadaten ohne

```
@prefix k4m1: <https://k4m.example/records/1#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <https://schema.org/> .

<https://k4m.example/records/1>
    a schema:Dataset ;
    rdfs:isDefinedBy [
        schema:Person [
            schema:givenName "Nico" ;
            k4m1:age [
                a schema:QuantitativeValue ;
                schema:value 28
            ]
        ]
    ] .
```

Abbildung 5.17: Auszug beispielhafter, als RDF exportierte und im Turtle-Format serialisierte Record-Metadaten.

Definition eines Terms wird stattdessen ein Kadi4Mat- und Record-spezifisches Präfix (in diesem Beispiel `k4m1`) definiert und verwendet.

Um letztendlich die Publizierung aller im Beispielexperiment beteiligten Objekte und Prozesse in Form der entsprechenden Collection durchführen zu können, kann die bereits in Kadi4Mat integrierte Zenodo-Schnittstelle verwendet werden. Der in diesem Kontext bereits erläuterte RO-Crate-Export, welcher bei der Publizierung zum Einsatz kommt, stellt dabei die strukturierte und gebündelte Bereitstellung sämtlicher Daten und Metadaten sicher, wobei die Wahl der zusätzlich enthaltenen Exportformate sowie weitere Inhalte direkt über die GUI oder HTTP-API von Kadi4Mat angepasst werden können.

6 Fallbeispiele

Im Laufe der Entwicklung wurde Kadi4Mat bereits in unterschiedlichen Fachbereichen der Ingenieurwissenschaften im Kontext mehrerer Forschungsprojekte und Kooperationen in der Praxis eingesetzt. Darunter fallen u. a. Anwendungen in den Materialwissenschaften [150, 190–192], inklusive Ansätze zur Laborautomatisierung [193], der Mikrostrukturtechnik [194], der Batterieforschung [195–198], dem Bioprinting [199, 200] sowie verschiedene, datenwissenschaftliche Einsatzgebiete [201–204]. Drei dieser Fallbeispiele werden in diesem Kapitel näher betrachtet. Der Fokus liegt dabei jeweils in der Umsetzung des FDMs unter Verwendung der verschiedenen von Kadi4Mat bereitgestellten Funktionalitäten und Schnittstellen. Pro Fallbeispiel wird zuerst der Einsatz von Kadi4Mat motiviert und anschließend dessen Verwendung im Kontext des konkreten Anwendungsfalls beschrieben und evaluiert. Aufgrund zeitlicher Unterschiede können sich die konkreten Funktionalitäten von Kadi4Mat je nach Fallbeispiel unterscheiden bzw. von dem in Kapitel 5 erläuterten Stand leicht abweichen.

6.1 Erzeugung FAIRer Forschungsdaten in der experimentellen Tribologie

In diesem Fallbeispiel wird mithilfe der von Kadi4Mat bereitgestellten sowie zusätzlich entwickelter Funktionalitäten gezeigt, wie die Erzeugung und Verwaltung FAIRer Forschungsdaten für ein tribologisches Experiment ermöglicht wird. Als Grundlage dient ein Vorzeigeexperiment typischen Aufbaus, bei dem ein geschmierter Stift (Gegenkörper) mit definierter Kraft auf eine rotierende Scheibe

(Grundkörper) gepresst wird. Dies geschieht mithilfe eines Tribometers, das ebenfalls im Stande ist, die resultierende Reibung und den Verschleiß zu messen. Das Vorzeigeexperiment ist absichtlich möglichst einfach gehalten, um den Fokus auf die Aspekte der Datengenerierung und -verwaltung zu legen, während gleichzeitig die Menge der entstehenden Daten und Metadaten überschaubar bleibt.

Das Experiment selbst wurde von einer Arbeitsgruppe am Institut für Angewandte Materialien (IAM) des KIT durchgeführt. Das Ziel der Kollaboration bestand darin, mithilfe von Kadi4Mat ein FAIRes Datenpaket zu erzeugen und zu veröffentlichen, das ein Muster für die Generierung von Datenpublikationen in der experimentellen Tribologie darstellen kann und sämtliche Rohdaten sowie verarbeitete Daten und Metadaten umfasst, sowohl in menschenlesbaren als auch in maschinenlesbaren Formaten. Im Hinblick auf die Nutzung und Entwicklung von Kadi4Mat liegt der Schwerpunkt dieses Fallbeispiels auf den dabei relevanten technischen Aspekten, wobei Teile von diesen ebenfalls in der entsprechenden Publikation von Brandt et al. [150] veröffentlicht sind. Dazu gehören einige zusätzliche Werkzeuge, die entwickelt wurden, um die Lücke zwischen den bestehenden technischen Lösungen in Form von Kadi4Mat und den spezifischen Anforderungen der Experimentatoren zu schließen. Alle weiteren Aspekte, einschließlich der Resultate des Vorzeigeexperiments, sind dagegen in [192] genauer beschrieben.

6.1.1 Motivation

Die experimentelle Tribologie, die sich mit der Beschreibung von unterschiedlichen, aufeinander einwirkenden Oberflächen befasst, stellt eine Teildisziplin der Materialwissenschaften dar, in der die Digitalisierung bisher vergleichsweise wenig fortgeschritten ist. Tribologie ist ein hoch interdisziplinäres Forschungsgebiet [205], das aufgrund des Mangels an allgemeingültigen Methoden für die Quantifizierung von Prozessen wie Reibung und Verschleiß zusätzliche Schwierigkeiten mit sich bringt. Speziell in der experimentellen Tribologie sind zudem die verwendeten Laborgeräte häufig maßgeschneidert und entsprechende Experimente werden oft „on the fly“ angepasst [192]. Entsprechend fehlt es derzeit an einer

disziplinspezifischen Forschungsdateninfrastruktur, welche die Aufzeichnung der gesamten Abfolge von Prozessen und äußeren Einflüssen in tribologischen Arbeitsabläufen und damit die Generierung FAIRer Daten unterstützt.

Die Notwendigkeit zur Digitalisierung tribologischer Experimente wird prinzipiell schon seit langem diskutiert [206, 207]. Bisherige Bemühungen waren insbesondere auf die Entwicklung von Datenbanken zur Sammlung und Wiederverwendung ausgewählter, tribologischer Kenngrößen fokussiert, wie sie z. B. durch die webbasierte Datenbank Tribocollect [208] in Form individueller, numerischer Datensätze bereitgestellt werden. Diese ist nach aktuellem Stand nach wie vor zugänglich, der Zugriff auf die vollständige Datenbank ist jedoch kostenpflichtig und es ist nicht eindeutig ersichtlich, ob das System und der Datenbestand weiterhin aktiv weiterentwickelt und gepflegt werden. Die Veröffentlichung tribologischer Daten, entweder als alleinstehender Datensatz oder als Ergänzung zu einer Textpublikation, ist in der Tribologie bisher ebenfalls nicht üblich [209]. Publikationen wie [210] stellen eines der wenigen Beispiele dar, allerdings sind die Metadaten und Beschreibungen für die durchgeführten Experimente überwiegend in textueller, menschenlesbarer Form gegeben und können daher nicht als vollständig interoperabel oder wiederverwendbar bezeichnet werden.

Ausgehend von den erläuterten Anforderungen wird deutlich, dass ein Datenaustausch zwischen Tribologen ohne eine vollständige semantische Beschreibung der Daten nicht möglich ist. Zwar existieren Metadatenschemata für übergeordnete Disziplinen wie den Materialwissenschaften [211] oder den computergestützten Ingenieurwissenschaften in Form von EngMeta, um jedoch die Besonderheiten tribologischer Arbeitsabläufe adäquat erfassen zu können, ist das von solchen Schemata bereitgestellte Vokabular in der Regel nicht spezifisch genug. Neben den Metadaten selbst muss zudem die Datenherkunft so präzise wie möglich beschrieben werden, um diese bis zum Ursprung verfolgen zu können, da auch scheinbar unbedeutende äußere Einflüsse einen signifikanten Einfluss auf die Ergebnisse eines bestimmten Experiments haben können [192].

Eine Möglichkeit, die erläuterten Anforderungen umsetzen zu können, stellen Ontologien und deren zugrunde liegende Terminologie und Beziehungen dar. Da

die Erstellung einer geeigneten Ontologie von Grund auf eine schwierige Aufgabe ist, selbst bei der Beschränkung auf eine bestimmte Disziplin, bauen bestehende Ontologien häufig auf Ontologien höherer Ebenen auf. Ein solcher Ansatz wird auch von der Ontologie tribAIn [212] verfolgt, die darauf abzielt, alle Arten von tribologischem Wissen in textbasierten Publikationen zu extrahieren. Dies erfordert jedoch ein hohes Maß an manueller Annotation und ist lediglich auf die innerhalb einer Publikation in natürlicher Sprache erläuterten Inhalte beschränkt. Um entsprechendes Wissen möglichst fehlerfrei und eindeutig formalisieren zu können, bieten sich daher zwei komplementäre Ansätze an: der Fokus einer entsprechenden Ontologie auf einen möglichst begrenzten Bereich, sowie die Produktion von Metadaten möglichst nah an deren Quelle, um die Vollständigkeit maximieren und manuell durchzuführende Arbeit bzw. potenzielle Fehlerquellen minimieren zu können.

6.1.2 Ergebnisse

Während sich dieses Fallbeispiel auf die technischen Aspekte der FAIRen Datenproduktion mit Kadi4Mat konzentriert, wurden von den Experimentatoren verschiedene Vorarbeiten geleistet, um die gesamte Datenproduktionspipeline zu realisieren, weshalb diese ebenfalls kurz vorgestellt werden. Eine Übersicht über die gesamte Pipeline ist in Abbildung 6.1 dargestellt, wobei die Vorarbeiten insbesondere den ersten Schritt der Digitalisierung der experimentellen Schritte umfassen, parallel zur Durchführung des eigentlichen Experiments. Als Teil dieser Vorarbeiten wurde zuerst ein kontrolliertes Vokabular erstellt, das sämtliche Objekte und Prozesse des tribologischen Experiments beschreibt. Hierdurch konnte sichergestellt werden, dass die innerhalb der Arbeitsgruppe verwendete Terminologie konsistent ist und einer zuvor definierten Semantik folgt. Zur kollaborativen Erstellung des Vokabulars kam eine lokal installierte Instanz der Software MediaWiki [213] zum Einsatz. Während die webbasierte Verwendung der Software die einfache Kollaboration ermöglicht und zusätzliche Funktionen wie z. B. Versionskontrolle bietet, ist die unstrukturierte Natur der eingetragenen Texte für die Weiterverarbeitung des Vokabulars weniger geeignet. Aus diesem Grund wurde

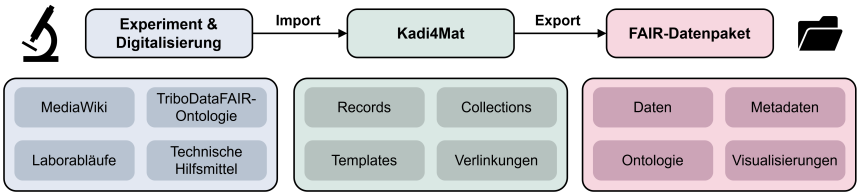

Abbildung 6.1: Visualisierung der Datenproduktionspipeline, beginnend mit dem eigentlichen Experiment und dessen Digitalisierung bis hin zum vollständigen FAIR-Datenpaket. Kadi4Mat fungiert als Brücke zwischen beiden Schritten, indem die strukturierte Speicherung von Daten und Metadaten ermöglicht und die notwendigen Import- und Exportfunktionalitäten bereitgestellt werden.

das entstehende Vokabular in eine entsprechende Ontologie transformiert, die mithilfe der Software Protégé [214] realisiert wurde, ein speziell auf die Modellierung von Ontologien zugeschnittener Editor. Die als TriboDataFAIR [215] bezeichnete und mithilfe von OWL beschriebene Ontologie bedient sich den übergeordneten Ontologien SUMO [216] und EXPO [217], die zur allgemeinen Beschreibung unterschiedlicher Konzepte sowie wissenschaftlicher Experimente eingesetzt werden können, legt jedoch den Fokus auf die FAIRe Produktion von Daten und Metadaten des Vorzeigeexperiments sowie auf das Experiment selbst. Ziel bei Erstellung der Ontologie war, deren Informationsgehalt so zu gestalten, dass die vollständige Reproduzierbarkeit des Experiments ausschließlich auf der Grundlage der Ontologie möglich ist [192].

Wie in Abbildung 6.1 dargestellt, fungiert Kadi4Mat hauptsächlich als Brücke zwischen dem Experiment und dem Export der entstehenden Daten als Teil des FAIR-Datenpakets. Um diese möglichst unkompliziert den Tribologen zur Verfügung stellen zu können, wurde für dieses Fallbeispiel eine bereits existierende, zentral am KIT gehostete Instanz von Kadi4Mat verwendet. Neben dem ersparten Installations- und Administrationsaufwand, wurde dadurch ebenfalls die Zusammenarbeit mit externen Forschern außerhalb der Arbeitsgruppe der Tribologen erleichtert. Für die Zusammenarbeit in der Arbeitsgruppe selbst wurde eine Gruppe mit allen beteiligten Experimentatoren innerhalb von Kadi4Mat angelegt, die je nach Bedarf mit entsprechenden Rollen ausgestattet werden konnte, um alle

im Laufe des Vorzeigeexperiments erstellten Ressourcen einsehen und bearbeiten zu können. Externen Beteiligten konnten hingegen individuelle Rollen mit unterschiedlichen Berechtigungen zugewiesen werden.

Aus Sicht von Kadi4Mat beginnt die eigentliche Datenproduktionspipeline mit dem Import aller entstehenden Daten und Metadaten. Dieser erfolgte sowohl in manueller als auch in automatisierter Form über die GUI bzw. die HTTP-API von Kadi4Mat. Durch die inhärent analoge Natur vieler Prozesse in der experimentellen Tribologie kam in diesem Fallbeispiel überwiegend erstere Form zum Einsatz. Ähnlich wie im Kapitel 5 gezeigten Beispielexperiment wurden sämtliche Prozessschritte, Geräte und Proben mithilfe von Records repräsentiert. Diese wurden innerhalb einer Collection gruppiert, welche das gesamte Experiment umfasst. Ein Beispiel eines solchen Records ist in Abbildung 6.2 dargestellt. Dieser Record stellt den für das Experiment vorbereiteten Gegenkörper dar, der durch verschiedene Aufbereitungsschritte aus dem Rohmaterial hergestellt wurde. Da es sich bei diesem um das digitale Gegenstück zu einem physischen Objekt handelt, besteht der entsprechende Record lediglich aus Metadaten. Die generischen Metadaten bilden dabei den Kern der Beschreibung der eigentlichen, experimentellen Entität und umfassen in diesem Fall allgemeine Informationen, sowie Angaben über die Dimensionen und das Material der Probe. Abbildung 6.3 zeigt den Record bzw. Gegenkörper im Zusammenhang des gesamten, experimentellen Arbeitsablaufs. Ein Großteil der für das Experiment erzeugten Records stellen die für die Vorbereitung der Proben notwendigen Prozesse dar, wie z. B. das Schleifen (z. B. *Cup Grinding #0003* in Abbildung 6.3) oder Säubern der Proben. Da es sich bei diesen um teilweise wiederkehrende Prozesse handelt, wurden ebenfalls entsprechende Templates erstellt und eingesetzt, um die innerhalb von Kadi4Mat repräsentierten Records effizienter erfassen zu können.

Sowohl für die Templates als auch zur davon unabhängigen Erstellung von Records wurde die TriboDataFAIR-Ontologie als Grundlage für die Inhalte und Struktur der grundlegenden sowie generischen Metadaten verwendet. Um die in der Ontologie enthaltenen Informationen nutzbar zu machen, wurde ein separates Werkzeug namens SurfTheOWL [218] entwickelt, welches im Kern auf der

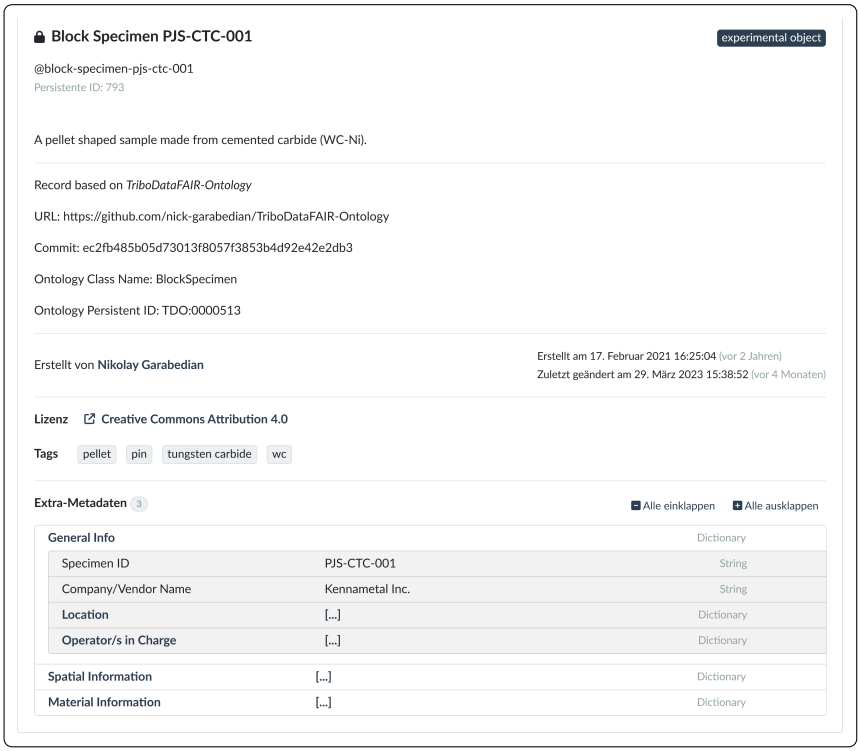

Abbildung 6.2: Screenshot eines Records in Kadi4Mat, welcher den für das Vorzeigeexperiment vorbereiteten Gegenkörper repräsentiert.

Python-Bibliothek Owlready2 [219] aufbaut und mithilfe des Webframeworks Django ebenfalls eine webbasierte GUI bereitstellt. Ähnlich wie Kadi4Mat selbst stellt SurfTheOWL eine Brücke dar, in diesem Fall zwischen den mithilfe der Ontologie formalisierten, jedoch vergleichsweise abstrakten, Objekten und Prozessen, sowie einer in der Praxis von Kadi4Mat nutzbaren Struktur. Die in der Ontologie spezifizierte Klassenhierarchie wurde dabei in eine passende „Metadatenhierarchie“ umstrukturiert. Mithilfe der GUI von SurfTheOWL konnte diese visualisiert werden und als Grundlage zur manuellen Eingabe von Metadaten dienen, während ein zusätzlich implementierter JSON-Export zur automatisierten Erstellung von Templates über die HTTP-API von Kadi4Mat verwendet wurde.

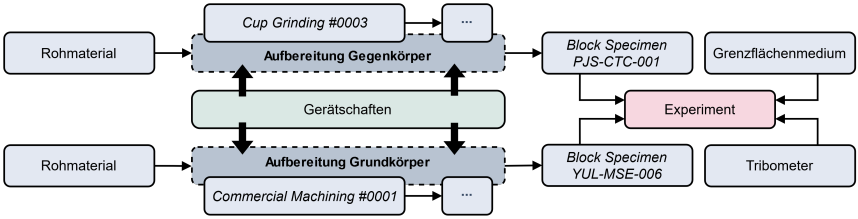

Abbildung 6.3: Vereinfachte Zeitleiste der am Vorzeigeexperiment beteiligten Objekte und Prozesse, beginnend mit den Rohmaterialien und endend mit dem eigentlichen Experiment, inklusive der verwendeten Proben, dem Tribometer sowie einem Schmiermittel als Grenzflächenmedium. Die im Experiment verwendeten Proben, Gegenkörper (*Block Specimen PJS-CTC-001*, dargestellt als Record in Abbildung 6.2) und Grundkörper (*Block Specimen YUL-MSE-006*), werden mithilfe unterschiedlicher Prozesse und unter Zuhilfenahme verschiedener Geräte zuvor passend aufbereitet. Eine detailliertere Version dieser Zeitleiste ist in [192] zu finden.

Auch wenn die Verwendung von Templates bereits dazu beitragen kann, Fehler bei der manuellen Eingabe von Metadaten zu reduzieren, sollte auch in diesem Fallbeispiel die automatische Erfassung von Metadaten möglichst nah an der Quelle die bevorzugte Methode darstellen. Da die von den Experimentatoren verwendeten Tribometer üblicherweise mithilfe der grafischen Entwicklungsumgebung LabView [220] gesteuert werden, bietet sich dieses Werkzeug als Schnittstelle für die automatisierte Datenerfassung der Tribometer an. Zusätzlich zu den bereits in der Entwicklungsumgebung eingebauten Funktionalitäten, ermöglicht LabView die Ausführung und Parametrisierung beliebiger Befehle über die Kommandozeile, sofern die entsprechenden Programme auf demselben Rechner wie LabView verfügbar sind. In diesem Kontext kam die CLI der Python-Bibliothek kadi-apy zum Einsatz, um innerhalb von LabView mit der HTTP-API von Kadi4Mat zu kommunizieren und einen passenden Record zu erstellen, welcher den computergestützten Prozess des eigentlichen, tribologischen Experiments beschreibt. Der entsprechende Befehl zur Erstellung des Records wurde dabei an das Ende eines bestehenden LabView-Workflows angefügt, während zusätzliche Daten und Metadaten dem Record im Laufe des Workflows über weitere Befehle hinzugefügt wurden. Abbildung 6.4 zeigt einen Ausschnitt dieses Workflows, welcher den erstgenannten Schritt umfasst. Die Konvertierung der Metadaten in das von der HTTP-API von Kadi4Mat erwartete Format konnte ebenfalls direkt mithilfe von

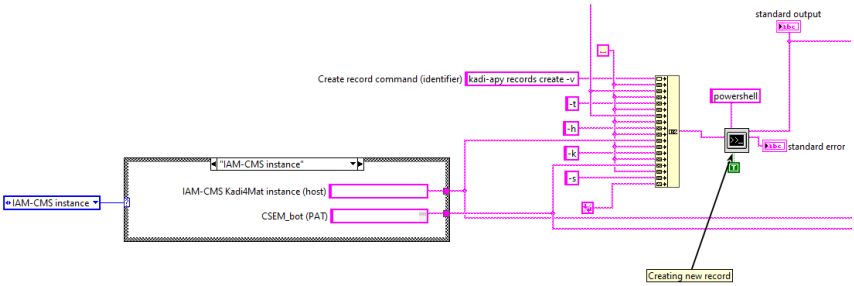

Abbildung 6.4: Ausschnitt aus einem LabView-Workflow, der einen von der Python-Bibliothek kadi-apy bereitgestellten Befehl zur Erstellung eines neuen Records zeigt. Dieser wird mithilfe von Windows PowerShell ausgeführt. Neben der Spezifikation des eigentlichen Befehls (`kadi-apy records create`) ist ebenfalls dessen Parametrisierung dargestellt, inklusive der Authentifizierung mit einer konfigurierten Instanz von Kadi4Mat über ein entsprechendes PAT.

LabView implementiert werden, weshalb sich diese Form der Integration ebenfalls um weitere Experimente und Tribometer erweitern lässt.

Während bisher hauptsächlich die Erstellung und Verwaltung individueller Records betrachtet wurde, die innerhalb einer einzigen Collection organisiert sind, war die Angabe der Beziehungen zwischen den Records ebenso wichtig. Für diese Zwecke bietet sich die Verwendung von Record-Links an, deren Records und Metadaten ebenfalls aus der TriboDataFAIR-Ontologie abgeleitet und manuell über die GUI von Kadi4Mat erstellt wurden. Beispielsweise wurden die beiden Records der in Abbildung 6.3 dargestellten Entitäten *Cup Grinding #0003* und *Block Specimen PJS-CTC-001* mithilfe der Beziehung *physicallyModifies* in Verbindung gesetzt, da der durch den erstgenannten Record repräsentierte Schleifprozess den Gegenkörper physisch verändert, um eine gleichmäßige Oberfläche zu gewährleisten. Eine Übersicht über diese und weitere den Gegenkörper betreffenden Verlinkungen sind in Abbildung 6.5 dargestellt, wobei es sich um dieselbe Art von Graph handelt, die bereits im Kontext des beispielhaften Anwendungsfalls in Kapitel 5 vorgestellt wurde. Da diese Art von Darstellung die logischen Beziehungen der Records umfasst, wird hierdurch bereits eine vereinfachte Form der zugrunde liegenden Ontologie sichtbar. Um die experimentellen Schritte ebenfalls in zeitlicher Anordnung visualisieren zu können, wurde eine weitere Übersicht

Abbildung 6.5: Visualisierung der mit dem Gegenkörper assoziierten, experimentellen Objekte und Prozesse in Form entsprechender Records und Record-Links. Der Gegenkörper des Vorzeigeexperiments wird durch den dunkelblauen Knoten im Zentrum des Graphen repräsentiert, während der zuvor bereits erwähnte Schleifprozess, welcher den Gegenkörper physisch verändert, als einer der zwei hellgrünen Knoten im unteren Teil des Graphen zu erkennen ist.

mithilfe eines zusätzlichen Python-Skripts [221] und der Visualisierungssoftware Graphviz [222] erstellt, die in Abbildung 6.6 dargestellt ist. Aufgrund der zeitlichen Anordnung ähnelt diese Art der Visualisierung stärker der in Abbildung 6.3 dargestellten Zeitleiste, wodurch eine Konsistenzprüfung der erstellten Records und Verlinkungen möglich wird. Zwar benötigt die Generierung einer solchen Visualisierung ein gewisses Maß an zusätzlichem Code, doch lassen sich auf diese Weise benutzerdefinierte Visualisierungen erstellen, die ebenfalls auf ähnliche Anwendungsfälle ausgeweitet werden können.

Um letztendlich anhand der erstellten Records das FAIR-Datenpaket zusammenstellen zu können, bietet sich die in Kadi4Mat integrierte Exportfunktionalität an. Diese wurde während der Kollaboration um zusätzliche Filtermöglichkeiten erweitert, um unterschiedliche Informationen und Metadaten in den exportierten Daten herausfiltern zu können, z. B. Benutzerinformationen, verlinkte Ressourcen oder individuelle, generische Metadaten. Ein Beispiel der Verwendung dieser

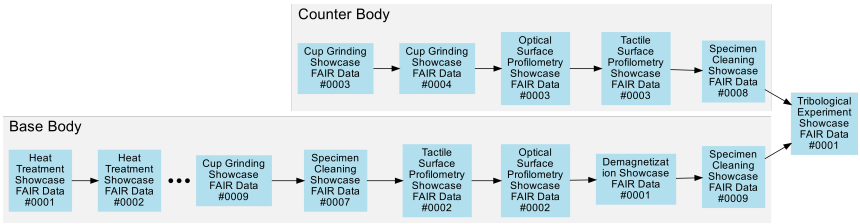

Abbildung 6.6: Automatisch generierte und gekürzte Visualisierung der wichtigsten tribologischen Proben, Gegenkörper (englisch: Counter Body) und Grundkörper (englisch: Base Body), sowie der mit ihrer Verarbeitung verbundenen Prozesse, bevor diese als Teil des eigentlichen Vorzeigeexperiments zum Einsatz kommen. Jede Probe wird in einer separaten Zeitleiste mit allen zugehörigen Prozessen angezeigt, geordnet nach dem Zeitstempel des jeweiligen Records. Die Zeitstempel geben den Zeitpunkt an, zu dem der jeweilige Prozess tatsächlich durchgeführt wurde, und sind im Gegensatz zum Erstellungsdatum der einzelnen Records als Teil der generischen Metadaten gespeichert.

Funktion innerhalb der GUI von Kadi4Mat ist in Abbildung 6.7 dargestellt. Für das FAIR-Datenpaket wurde diese Funktion hauptsächlich dazu verwendet, um sowohl automatisch erfasste als auch in den generischen Metadaten enthaltene, benutzerbezogene Informationen der Records auszuschließen. Aufgrund der zum Zeitpunkt der Durchführung dieses Fallbeispiels bestehenden Beschränkung der Exportfunktionalität auf einzelne Records, wurde von der HTTP-API von Kadi4Mat Gebrauch gemacht, um alle Records innerhalb der für das Experiment verwendeten Collection für das FAIR-Datenpaket gezielt exportieren und bündeln zu können. Hierzu kam ebenfalls ein zusätzlich entwickeltes Python-Skript [221] zum Einsatz. Neben den eigentlichen Daten wurden pro Record dessen Metadaten im JSON- sowie im PDF-Format exportiert, um eine maschinen- als auch menschenlesbare Form der enthaltenen Informationen bereitzustellen. Das gesamte Datenpaket wurde anschließend, inklusive zusätzlicher Visualisierungen, im Repositorium Zenodo veröffentlicht [223].

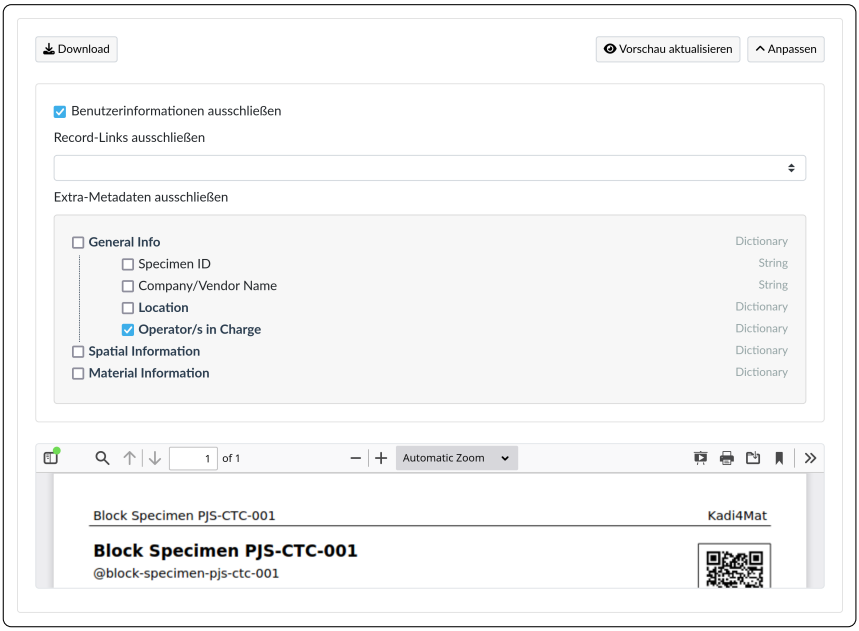

Abbildung 6.7: Screenshot der Filtermaske eines Records beim Export im PDF-Format. Der Record stellt auch hier den aufbereiteten Gegenkörper dar, der bereits in Abbildung 6.2 dargestellt ist. In diesem Beispiel werden die allgemeinen Benutzerinformationen, die verknüpften Collections und ein bestimmtes Metadatum in der generierten PDF-Datei ausgeschlossen.

6.1.3 Fazit

In der Kollaboration zwischen Entwicklern und Experimentatoren wurde deutlich, welche Vorteile es bietet, Systeme wie Kadi4Mat parallel zur Durchführung der eigentlichen Forschungsarbeit zu nutzen und weiterzuentwickeln. Auf diese Weise konnten zusätzlich entwickelte Funktionen zeitnah in der Praxis umgesetzt und durch entsprechendes Feedback evaluiert und verbessert werden. Die experimentelle Tribologie stellt aufgrund der Anforderungen an maßgeschneiderte und sich dennoch ständig ändernde, interdisziplinäre Arbeitsabläufe einen idealen Anwendungsfall dar, um generische und flexible Funktionalitäten umsetzen zu können. Dieser Aspekt entspricht ebenfalls den der Entwicklung von Kadi4Mat

zugrunde liegenden Konzepten und lässt sich leicht auf andere Forschungsbereiche übertragen.

Wie bereits erwähnt, kann Kadi4Mat innerhalb der Datenproduktionspipeline als Brücke zwischen der Durchführung des Experiments sowie dem resultierenden Datenpaket betrachtet werden. Neben der Durchführung des Vorzeigeexperiments selbst waren jedoch unterschiedliche Vorarbeiten zur Digitalisierung der experimentellen Schritte notwendig, einschließlich der Definition von Begriffen zur Erstellung eines kontrollierten Vokabulars als Grundlage der TriboDataFAIR-Ontologie. Top-down-Ansätze wie diese erfordern nicht nur viel Arbeit für die Domänenexperten, in diesem Fall die Tribologen, sondern auch interdisziplinäres Wissen in anderen Bereichen als der jeweiligen Domäne, selbst bei Beschränkung auf z. B. eine bestimmte Art von Experiment. Um die Hürden zur Etablierung solcher Arbeitsabläufe möglichst gering zu halten, sollte ebenfalls die Notwendigkeit für Vorarbeiten sowie zur Entwicklung zusätzlicher Werkzeuge gering gehalten werden. Da durch die Flexibilität von Kadi4Mat die Organisation der Daten und Metadaten größtenteils offengehalten wird, besteht insbesondere bei der Strukturierung von Metadaten Unterstützungsbedarf, sofern einheitliche Schemata verwendet werden sollen, jedoch keine passenden, existierenden Schemata zur Verfügung stehen. Die im Kontext der Kollaboration verwendeten und teilweise speziell entwickelten Werkzeuge, wie MediaWiki, Protégé und SurfTheOWL, können bei der Erstellung passender Schemata helfen, sind jedoch größtenteils maßgeschneidert. Eine generische Alternative stellt VocPopuli [224] dar, ein auf Basis der Kollaboration entstandenes, webbasiertes Werkzeug zur kollaborativen Erstellung von Vokabularen. Diese können anschließend unter Verwendung der HTTP-API von Kadi4Mat direkt in Form entsprechender, generischer Metadaten-Templates exportiert werden, wobei das zugrunde liegende Vokabular in Form passender IRIs spezifiziert wird. Deren Angabe wird seit Durchführung dieses Fallbeispiels als Teil der generischen Record-Metadaten von Kadi4Mat unterstützt. VocPopuli kann damit als ein möglicher Startpunkt als Teil einer generischen und disziplinunabhängigen FAIR-Datenproduktionspipeline betrachtet werden und potenziell ebenfalls in Form eines Terminologieservices integriert werden.

Als Teil des FAIR-Datenpakets konnten alle Daten und Metadaten, die während der Durchführung des Experiments erstellt und gesammelt wurden, öffentlich bereitgestellt werden. Die einzige Ausnahme stellen von externen Lieferanten bereitgestellte Metadaten dar, etwa für Proben oder Geräte, die nicht in allen Fällen vollständig sind. Um die FAIRness des Datenpakets, insbesondere im Hinblick auf dessen Nachnutzung, qualitativ evaluieren zu können, wurde von den bereits erwähnten FAIRsFAIR Data Object Assessment Metrics Gebrauch gemacht, wobei eine detaillierte Auflistung der Metriken im Anhang von [150] zu finden ist. Durch die Nutzung eines öffentlichen Repositoriums wie Zenodo wird bereits ein Großteil dieser Metriken erfüllt, wie z. B. die Vergabe eines PIDs, die Auffindbarkeit des Datenpakets sowie dessen Nutzbarkeit über standardisierte Schnittstellen. Zwar liegt den exportierten Metadaten mangels existierender Schemata kein innerhalb der hier relevanten Forschungsgemeinschaft etablierter Standard zugrunde, durch die Verwendung eines kontrollierten Vokabulars, das auf existierenden Standards aufbaut, wird eine semantische Interoperabilität dennoch größtenteils ermöglicht. Insbesondere durch die auf dem Vokabular aufbauende Ontologie umfasst dieser Aspekt ebenfalls die Beziehungen der Daten und Metadaten untereinander sowie die dadurch definierte Datenherkunft.

Dennoch besteht Potenzial, um mithilfe von Kadi4Mat die FAIRness des Datenpakets zu verbessern, vor allem im Hinblick auf die Maschinenlesbarkeit der enthaltenen Daten und Metadaten. Da das Datenpaket selbst sowie die exportierten Metadaten keiner formalisierten Struktur folgen, wird eine automatisierte Extraktion und Interpretation der Inhalte erschwert. Eine mögliche Optimierung stellt daher die Verwendung der in Kapitel 5 bereits vorgestellten RO-Crates dar, deren Export seit Durchführung dieses Fallbeispiels in Kadi4Mat implementiert wurde. Diese können in Kombination mit dem ebenfalls seitdem implementierten, RDF-basierten Export von Metadaten zu einer verstärkten Interoperabilität sowohl auf syntaktischer als auch semantischer Ebene beitragen.

6.2 Datengetriebene Prozessüberwachung und -optimierung im Bioprinting

In diesem Fallbeispiel wird gezeigt, wie Kadi4Mat dazu beiträgt, sowohl experimentelle als auch computergestützte Prozesse im Rahmen eines Ringversuchs im Bereich des Bioprintings zu unterstützen. Als Grundlage des Fallbeispiels dient das Forschungsprojekt SOP_BioPrint, das zum Ziel hat, die Robustheit und Übertragbarkeit des extrusionsbasierten Bioprinting-Prozesses zu analysieren. Dafür wurde eine Infrastruktur zur Standardisierung und Kollaboration aufgebaut und validiert, bei der Kadi4Mat eine zentrale Rolle einnimmt. Die Infrastruktur umfasst die Entwicklung und systematische Dokumentation standardisierter Vorgehen (englisch: Standard Operating Procedures, kurz SOPs) und den Austausch von Daten und Metadaten für deren automatisierte Auswertung über verschiedene Standorte hinweg.

Als einheitlicher Prozess wurde eine Reihe von zellfreien Hydrogelstrukturen von insgesamt zwölf teilnehmenden Laboren gedruckt, inklusive der vorherigen Charakterisierung der Rohmaterialien bis hin zur unabhängigen Analyse der gedruckten Objekte. Alle innerhalb dieses Ringversuchs erzeugten SOPs und Daten wurden mithilfe von Kadi4Mat dokumentiert, organisiert und ausgetauscht, mit dem Ziel, die Reproduzierbarkeit zwischen den verschiedenen Standorten mit den jeweils lokal verfügbaren Gerätschaften zu ermöglichen. Für die Analyse anhand anonymisierter Bilddaten wurden zusätzliche Python-Skripte zur automatisierten Datenverarbeitung durch die beteiligten Analysegruppen entwickelt. Wie bereits beim ersten Fallbeispiel liegt der Fokus auf der Vorbereitung und Nutzung von Kadi4Mat in Bezug auf den technischen Umfang und den umgesetzten, digitalen Arbeitsfluss. Teile dieser Aspekte sind ebenfalls in der entsprechenden Publikation von Schmieg et al. [199] veröffentlicht, während die Durchführung und Evaluation des eigentlichen Ringversuchs in [200] näher beschrieben ist.

6.2.1 Motivation

Beim Bioprinting handelt es sich um eine vielversprechende Herstellungsmethode für die bedarfsorientierte Produktion maßgeschneiderter, organischer Objekte, bei der eine spezielle Form des 3D-Drucks zum Einsatz kommt. Das Verfahren erzeugt dreidimensionale Strukturen aus biokompatiblem Material und eingebetteten Zellen. Die Einsatzgebiete vom Bioprinting liegen sowohl in der pharmazeutischen Forschung, z. B. bei der Durchführung zellbasierter Screenings [225], als auch im medizinischen Bereich, wo die Herstellung von patientenspezifischem Gewebeersatz mit komplexen Geometrien möglich ist [226]. Die allgemeine Durchführung entsprechender Druckprozesse umfasst die Vorbereitung des zu druckenden Materials, auch als Biotinte bezeichnet, der eigentliche Druckvorgang sowie verschiedene Formen der Nachbearbeitung [227]. Durch die unterschiedlichen Arten von anwendungsabhängigen Biotinten und Drucktechnologien [228], sowie der dezentralen und bedarfsgerechten Produktion der zu druckenden Objekte, entsteht eine hohe Anzahl an miteinander verbundenen Variablen, die das Druckerzeugnis beeinflussen können [229].

Im Vergleich zu anderen Bereichen, in denen 3D-gedruckte Objekte bzw. additive Fertigung zum Einsatz kommen, stellt die Einzigartigkeit des Bioprintings eine vergleichsweise hohe Hürde für den Markteintritt in Kliniken dar. Während in Disziplinen wie dem Maschinenbau die Qualität 3D-gedruckter Komponenten mit durch etablierte Verfahren hergestellten, äquivalenten Objekten in Hinsicht auf Geometrie und Funktionalität verglichen werden kann, ist dies beim Bioprinting aufgrund der stark anwendungsspezifischen Natur des Verfahrens nur schwer möglich. Entsprechendes gilt ebenfalls bei der Etablierung von standardisierten Druckprozessen und Protokollen zur Qualitätskontrolle [230]. Zudem können beim Bioprinting bestimmte Objekte potenziell lediglich für einen einzigen Patienten entworfen werden, was zusätzliche regulatorische Verfahren zur Bewertung der Sicherheit und Wirksamkeit im Bereich der personalisierten Medizin erfordern kann [231].

Unterschiedliche Aspekte, die bestimmte Herstellungsparameter mit dem Ergebnis des 3D-Drucks verbinden, wurden bereits in Isolation untersucht, wie z. B. das rheologische Verhalten der Biotinte und deren Einfluss auf die Dicke der gedruckten Stränge [232] oder Bildanalysen zur Erkennung von Abweichungen zwischen dem zu druckenden Modell und dem eigentlichen Druckerzeugnis [233]. Um jedoch Standards zu erfüllen, wie sie bereits in der Pharmaindustrie etabliert sind [234], und damit ebenfalls den Transfer in die Industrie zu ermöglichen, sind Aspekte wie die Sicherheit, Reproduzierbarkeit und Qualitätskontrolle zu berücksichtigen [231]. Um diese bereits während der Planungs- und frühen Entwicklungsphase eines Druckvorgangs im Bioprinting gewährleisten zu können, besteht daher ein Bedarf an einem optimierten und vor allem datengetriebenen Prozessverständnis [235–237], wodurch ebenfalls Potenzial zur Anwendung von Data-Science-Algorithmen eröffnet wird [238]. Dieses Verständnis setzt eine umfassende und lückenlose Dokumentation des Herstellungsprozesses voraus, was nur durch eine vollständige Digitalisierung aller Arbeitsschritte ermöglicht werden kann. Dazu gehört die systematische und strukturierte Dokumentation der Experimente in einem ELN oder vergleichbaren System, die digitale Bereitstellung von SOPs sowie ein möglichst automatisierter Datentransfer an Analysewerkzeuge. Diese Anforderungen müssen von einer entsprechenden Forschungsdateninfrastruktur unterstützt werden, wobei es derzeit an geeigneten Lösungen zur Verwendung im Bioprinting mangelt.

6.2.2 Ergebnisse

Um einen erfolgreichen Ringversuch sowie dessen Digitalisierung mit den lokal verfügbaren Geräten der verschiedenen Standorte gewährleisten zu können, musste eine gewisse Kontrolle und Überwachung über den gesamten Prozess möglich sein. Dies ist nicht nur dem interdisziplinären Charakter des Bioprintings geschuldet, sondern auch dem unterschiedlichen Vorwissen der Forscher zur effizienten Arbeit mit Forschungsdateninfrastrukturen. Die Prozessschritte des Ringversuchs wurden daher auf der Grundlage von SOPs und standardisierten Templates für die Erfassung von Daten und Metadaten durchgeführt, die in Kadi4Mat digital

zur Verfügung gestellt wurden. Hierzu kam ebenfalls die bereits im vorherigen Fallbeispiel verwendete, zentral am KIT gehostete Instanz von Kadi4Mat sowie unterschiedliche Benutzergruppen zum Verwalten von Zugriffsrechten zum Einsatz. Um den Gestaltungsraum für die Betriebsparameter, SOPs und Templates auf Grundlage einer vorherigen Risikoanalyse sowie existierender Literatur zu definieren, wurden Vortests in den jeweiligen Laboren durchgeführt. Dadurch konnten die sogenannten kritischen Prozessparameter identifiziert werden, wie z. B. die Temperatur der Biotinte, die direkt mit deren Viskosität zusammenhängt [229] und sich wiederum in den darauffolgenden Schritten des Bioprinting-Prozesses auf Aspekte wie den Extrusionsfluss auswirken kann.

Der gesamte Ablauf eines einzelnen Experiments des Ringversuchs ist schematisch in Abbildung 6.8 dargestellt und teilt sich in verschiedene, hierarchisch organisierte Phasen auf. Die zuvor von der Projektadministration erstellten SOPs, sowie der Entwurf der zu druckenden Geometrien, sind dabei Teil der Vorverarbeitungsphase, um eine vergleichbare Ausführung der Prozesse zu gewährleisten. Die innerhalb der SOPs definierten Schritte wurden entweder als „standardisierte Handlungen“ oder als Ausführung „unter Berücksichtigung der örtlichen Gegebenheiten innerhalb eines Parameterfensters“ angegeben. Beispielsweise konnte der Entwurf der zu druckenden Geometrien problemlos zwischen verschiedenen Standorten übertragen werden, während die Einstellungen der lokalen Bioprinting-Ausrüstung je nach Umgebungsbedingungen, Standort oder in Abhängigkeit von der Zeit variieren. Ebenso konnte die Überwachung der Materialien zur Herstellung der Biotinten zentral vom jeweiligen Lieferanten durchgeführt werden, während die Vorbereitung der Biotinten und der eigentliche Druckprozess in den täglichen Arbeitsablauf der unterschiedlichen Labore integriert wurden.

Für die einzelnen Labore spielt innerhalb der ersten Phase des in Abbildung 6.8 dargestellten Ablaufs hauptsächlich die Vorbereitung der entsprechenden Biotinten eine Rolle. Ein Beispiel eines in Kadi4Mat hinterlegten Templates, das von der Projektadministration zur standardisierten Dokumentation dieses Prozesses bereitgestellt wurde, ist in Abbildung 6.9 dargestellt. Neben den eigentlichen Inhalten

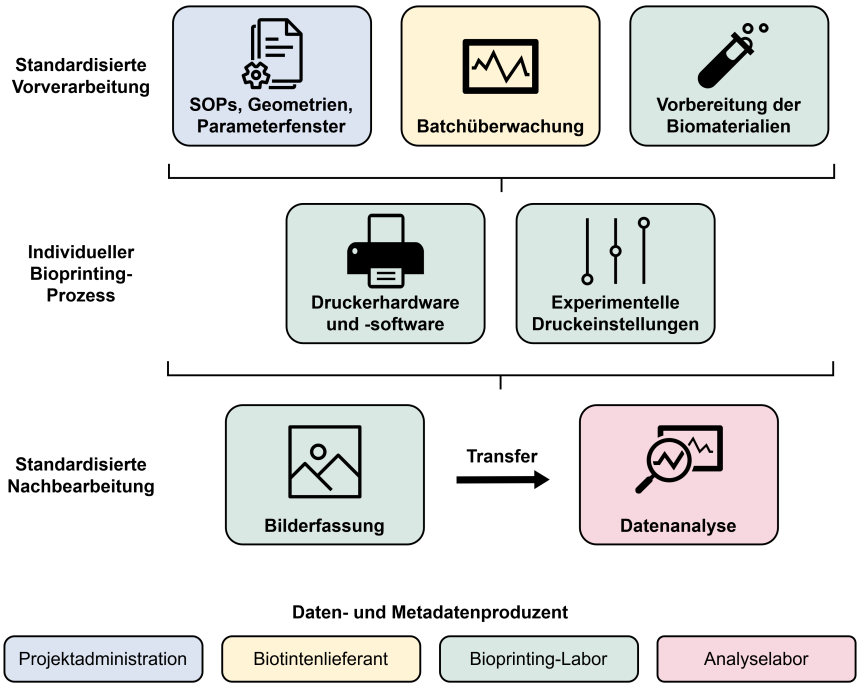

Abbildung 6.8: Schematischer Arbeitsablauf eines Bioprinting-Experiments und der damit verbundenen Daten und Metadaten. Der Prozess lässt sich grob in die Vorverarbeitungsphase, den eigentlichen Druckvorgang und die Nachbearbeitung unterteilen, welche ebenfalls die Analyse der Druckergebnisse in Form entsprechender Bilddaten umfasst. Die unterschiedlichen Farbcodes verdeutlichen, dass Daten und Metadaten der einzelnen Prozessschritte von allen beteiligten Akteuren erzeugt werden.

wurde die Beschreibung des Templates dazu verwendet, den Anwendern zusätzliche Informationen zu vermitteln und auf relevante SOPs zu verweisen. Ähnlich wurde die Dokumentation der zwei folgenden Phasen durchgeführt, welche den eigentlichen Druckprozess sowie die Nachbearbeitung und Qualitätskontrolle der Druckerzeugnisse umfassen. Um im Kontext letzterer eine objektive und quantitative Evaluation zu ermöglichen, wurden Bilddaten der gedruckten Objekte mithilfe eines speziell dafür entwickelten Gerätes standardisiert erfasst [200]. Sämtliche Bilder wurden anschließend zur unabhängigen und doppelblinden Analyse der Gesamtergebnisse insgesamt drei unterschiedlichen Laboren bereitgestellt. Um

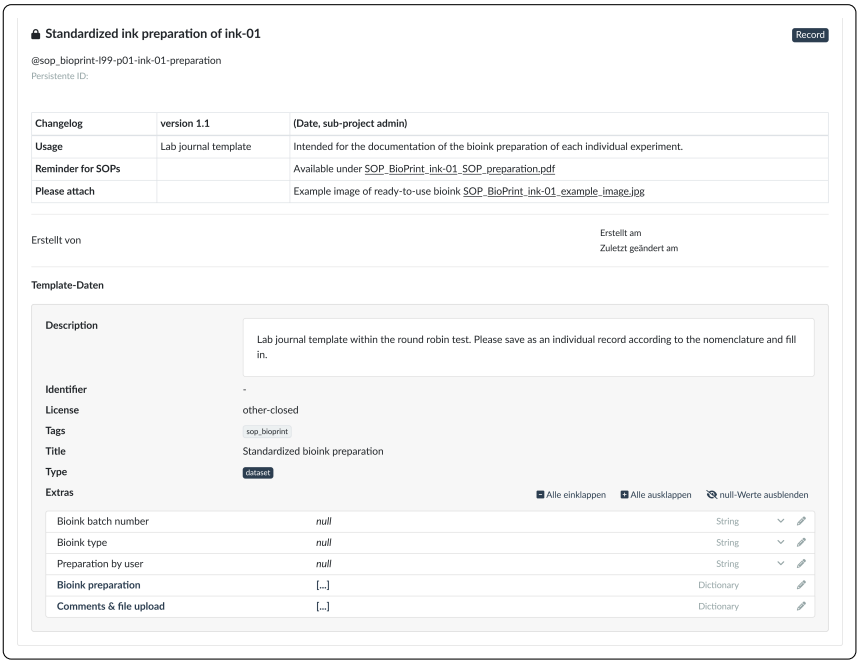

Abbildung 6.9: Screenshot eines Record-Templates in Kadi4Mat für die standardisierte Dokumentation der Vorbereitung der verwendeten Biotinten, wobei personenbezogene Daten entfernt wurden. Die Beschreibung des Templates im oberen Teil der Abbildung zeigt die Möglichkeit auf, nutzerlesbare Anweisungen spezifizieren zu können, während durch die im Template spezifizierten Metadaten der Vorbereitungsprozess detailliert erfasst werden kann. Hierbei wurden insbesondere die zusätzlichen Validierungsanweisungen der generischen Metadaten als Unterstützung benutzt, um bestimmte Wertebereiche vorgeben zu können.

die Analyse möglichst automatisiert ermöglichen zu können, kam in diesem Fallbeispiel erneut ein separat entwickeltes Python-Werkzeug zum Einsatz, das als Brücke zwischen der HTTP-API von Kadi4Mat und der anwendungsspezifischen Analysesoftware diente. Das Werkzeug wurde auf Basis der Python-Bibliothek kadi-apy entwickelt und um eine einfache GUI zur leichteren Bedienung erweitert, die zusätzlich zur direkten Nutzung über Python verwendet werden konnte und ebenfalls als Teil eines SOPs beschrieben wurde. Die zu transferierenden Daten

wurden zuvor von der Projektadministration anonymisiert und innerhalb unterschiedlicher Collections innerhalb von Kadi4Mat bereitgestellt. Zusätzlich zu den eigentlichen Daten ermöglicht das Werkzeug den Export sämtlicher Record-Metadaten in maschinen- und nutzerlesbaren Formaten, wodurch ein gezielter und automatisierbarer Transfer sichergestellt werden konnte.

Während die Analyse der Daten größtenteils automatisiert werden konnte, war ebenfalls die manuelle Überprüfung der anhand der SOPs und Templates erzeugten Ressourcen nützlich. Hierzu kamen auch in diesem Fallbeispiel Record-Links zum Einsatz, um mithilfe passender Verlinkungen zwischen den SOPs, Datensätzen und Druckern die Datenherkunft zu spezifizieren. Die entsprechenden, mithilfe von Kadi4Mat generierbaren Visualisierungen (vgl. Abbildung 6.5) konnten insbesondere zur manuellen Kontrolle des Dokumentationsfortschritts verwendet werden, sowohl von den individuellen Laboren als auch von der Projektadministration. Eine als Teil der GUI von Kadi4Mat bereitgestellte Vorschau unterschiedlicher Datenformate konnte genutzt werden, um den Überblick über die SOPs und hochgeladenen Datensätze während der eigentlichen Laborarbeit zu behalten. Diese wurde um zusätzliche Formate erweitert, u. a. zur Visualisierung von STL- und TIFF-Dateien, wie in Abbildung 6.10 und Abbildung 6.11 dargestellt. Während ersteres Format zum Austausch der zu druckenden Geometrien verwendet wurde, stellten TIFF-Dateien das primäre Datenformat zur Speicherung der zur Analyse verwendeten Bilddaten dar.

6.2.3 Fazit

Die flexible Struktur von Kadi4Mat war innerhalb dieses Fallbeispiels essenziell zur Umsetzung und Dokumentation des zu Beginn des Projekts definierten und kontinuierlich angepassten Arbeitsablaufs, was insbesondere die Spezifikation anwendungsspezifischer Templates und entsprechender Ressourcen umfasst. Bei der Durchführung des Ringversuchs lag der Schwerpunkt weniger auf der FAIRness der erzeugten Daten und Metadaten, sondern hauptsächlich auf der projektinternen Kollaboration zwischen den teilnehmenden Laboren unter Anleitung von in

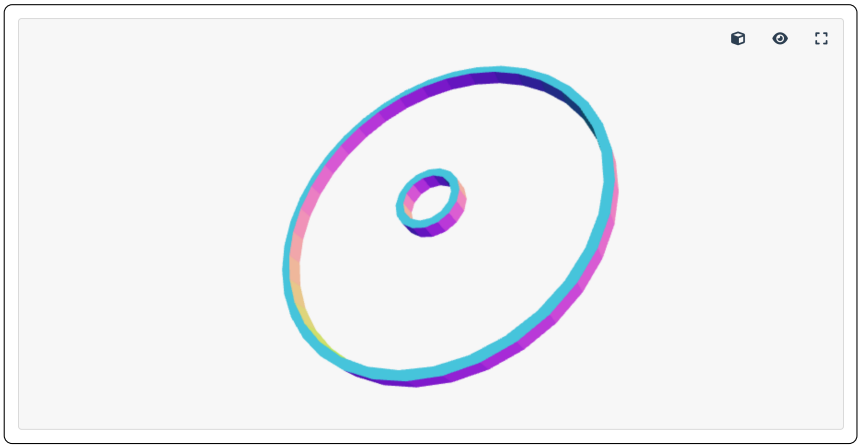

Abbildung 6.10: Screenshot der webbasierten Vorschau einer STL-Datei innerhalb von Kadi4Mat. Die Vorschau erlaubt eine interaktive Visualisierung, die mithilfe von Mausgesten beeinflusst werden kann.

Kadi4Mat hinterlegten SOPs in unterschiedlicher Form. Die feingranulare Verwaltung von Zugriffsrechten innerhalb von Kadi4Mat spielte eine wichtige Rolle, um nicht nur die gemeinsame Nutzung von Daten zu ermöglichen, sondern auch die Sichtbarkeit und Zugriffsrechte für bestimmte Gruppen einzuschränken. Dies war ebenfalls für die doppelblinde und anonymisierte Übergabe der Datensätze an die Prozessanalyse wichtig, sodass lediglich die Projektadministration auf alle hierarchisch gespeicherten Daten aus verschiedenen Laboren zugreifen konnte. Wie auch im vorherigen Fallbeispiel wurde deutlich, dass die Ermittlung neuer Anforderungen an ein System wie Kadi4Mat durch eine enge Zusammenarbeit zwischen Entwicklern und Nutzern bzw. der Projektadministration begünstigt wird.

Die Parameter, welche das Ergebnis des Bioprintings beeinflussen können, sind vielfältig und stark miteinander verknüpft, weshalb die systematische Prozessüberwachung mithilfe entsprechender Systeme und deren Integration mit prozessanalytischen Werkzeugen einen wichtigen Aspekt beim Transfer in die industrielle Anwendung darstellt. Gleichzeitig spielt jedoch auch die Arbeitseffizienz des Anwenders im Vergleich zu traditionellen, papierbasierten Laborbüchern eine

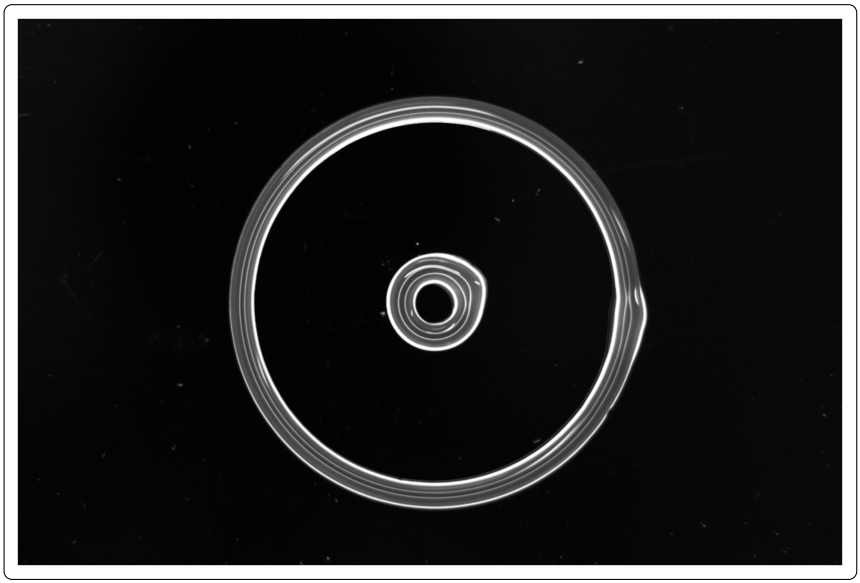

Abbildung 6.11: Screenshot der webbasierten Vorschau einer TIFF-Datei innerhalb von Kadi4Mat. Das dargestellte Bild zeigt eine Aufnahme eines konkreten Druckerzeugnisses, basierend auf der in Abbildung 6.10 dargestellten Geometrie.

Rolle. Abgesehen von der Bildanalyse, die mithilfe entsprechender Software durchgeführt wurde, waren für die restlichen Teilnehmer des Ringversuchs keine Programmierkenntnisse erforderlich. Insbesondere wurde die Möglichkeit, digitale Einträge zu kopieren, von den Nutzern als zeitsparend im Vergleich zu klassischen Laborbüchern empfunden. Die Einarbeitung in die Verwendung von Kadi4Mat und der vordefinierten Templates dauerte maximal eine Stunde und wurde von der Projektadministration begleitet. Aufgrund der unterschiedlichen Erfahrungen und Vorkenntnisse der Experimentatoren im Umgang mit Systemen wie Kadi4Mat war die Bedienung für einige Benutzer selbstverständlich, für andere schwieriger, was der Beteiligung interdisziplinärer Teams aus Biologie, Materialwissenschaften und Datenwissenschaften an verschiedenen Standorten geschuldet war.

Als Teil des Fallbeispiels wurden nicht nur individuelle Druckprozesse, sondern die gesamte Wertschöpfungskette eines möglichen Bioprinting-Prozesses betrachtet. Dieses Fallbeispiel kann daher als vergleichsweise einfach umzusetzende und stark praxisorientierte Vorlage für ähnliche Anwendungen sowohl im Bioprinting als auch anderen Disziplinen dienen, um digitalisierte und robuste Prozessketten zu entwickeln. Dennoch sind weitere Arbeiten erforderlich, um die beschriebenen Arbeitsabläufe zu optimieren, zusätzliche Anforderungen an Systeme wie Kadi4Mat zu identifizieren und letztendlich die Steigerung der Effizienz durch den Einsatz solcher Systeme bewerten zu können. Durch den Fokus auf manuelle Arbeitsschritte stellen insbesondere Funktionalitäten wie das Hinterlegen anwendungsspezifischer Anleitungen, integrierte Hilfestellungen sowie eine Erweiterung der vorhandenen Template-Funktionalität eine wichtige Entwicklungsperspektive für Kadi4Mat dar.

6.3 Automatisierte Arbeitsflüsse in der Mikrostrukturtechnik

In diesem Fallbeispiel wird die Nutzung von Kadi4Mat innerhalb automatisierter Arbeitsflüsse erläutert, die zur strukturierten Dokumentation und Digitalisierung von Prozessschritten in der Mikrostrukturtechnik zum Einsatz kommen. Die hierbei relevanten Arbeitsflüsse werden in den folgenden Abschnitten primär als Workflow bezeichnet, wobei es sich um die bereits in Kapitel 2 etablierte Definition einer wohldefinierten Abfolge von sequentiellen oder parallelen Schritten handelt, die so automatisch wie möglich abgearbeitet werden. Als konkreter Arbeitsablauf wird die Produktion von Mikrostrukturbauteilen aus Siliziumwafern betrachtet, die u. a. zur Herstellung von Demonstratoren am Institut für Mikrostrukturtechnik (IMT) des KIT dient [194]. Diese können z. B. zur Verifikation von Forschungsergebnissen oder im Rahmen wissenschaftlicher Arbeiten und Projekte eingesetzt werden. Bei der Produktion solcher Bauteile kommen sowohl

standardisierte als auch noch zu standardisierende Arbeitsschritte in festgelegter Reihenfolge zum Einsatz, die sich daher gut zur Umsetzung innerhalb eines Workflows eignen.

Aufgrund der generischen Funktionsweise von Workflows können diese in nahezu allen Anwendungsgebieten ihren Einsatz finden und sowohl simulative als auch experimentelle Arbeitsabläufe umfassen, weshalb die innerhalb eines Workflows definierten Schritte möglichst flexibel implementierbar sein müssen. Aus Benutzersicht setzt dies die einfache Bereitstellung entsprechender Werkzeuge voraus, deren Ausführung wiederum in einer funktionalen Umgebung erfolgen muss. Dies gilt insbesondere für die Verwendung bestehender Werkzeuge, z. B. einem in MATLAB [239] geschriebenen Skript, das zur korrekten Funktionsweise nicht nur eine MATLAB-Installation auf dem auszuführenden System benötigt, sondern ebenfalls mit einer geeigneten Schnittstelle zur Verwendung innerhalb eines Workflows ausgestattet sein muss. Diese Aspekte, sowie die Funktionsweise eines entsprechenden Workflows, werden als Teil des Fallbeispiels näher betrachtet. Der Schwerpunkt liegt dabei auf dem technischen Hintergrund des implementierten Workflows sowie in der Integration von Kadi4Mat unter Verwendung zusätzlicher Werkzeuge. Details zu den Konzepten, welche den Workflows und ihrer Ausführung im Allgemeinen zugrunde liegen, sind ebenfalls in den Publikationen von Griem et al. [240] und Brandt et al. [166] veröffentlicht.

6.3.1 Motivation

Bei der Produktion von in diesem Fallbeispiel betrachteten Siliziumwafern spielt die Standardisierung der dazu notwendigen Arbeitsschritte eine wichtige Rolle, da die entsprechenden Bauteile anfällig sind gegenüber falschen Prozessparametern, was ebenfalls Aspekte wie z. B. die inkorrekte Bedienung von Gerätschaften umfasst. Für diese Zwecke kommen bei der Fertigung von Werkstücken häufig sogenannte Laufkarten zum Einsatz. Bei diesen handelt es sich klassischerweise um physische Dokumente, welche den Ablauf der Herstellung und der Prüfung von Werkstücken definieren. Jeder Schritt ist bei dessen Abarbeitung typischerweise

mit zusätzlichen Metadaten zu kennzeichnen, um z. B. mögliche Abweichungen zum Ablauf oder Informationen zur durchführenden Person zu hinterlegen. Am IMT des KIT kommen unterschiedliche Arten von Laufkarten zum Einsatz, die durch verschiedene Farbcodes gekennzeichnet werden. Grüne Laufkarten stehen z. B. für einen fest definierten Arbeitsablauf, während blaue Laufkarten lediglich eine Vorstufe von diesen darstellen und unterschiedliche Abläufe für die einzelnen Prozessschritte enthalten können, die rückwirkend in Form einer grünen Laufkarte standardisiert werden können.

Bei dem hier betrachteten Workflow kommt der Ablauf einer blauen, also noch zu standardisierenden, Laufkarte zum Einsatz. Die einzelnen Schritte beinhalten Prozesse wie das Belichten oder Beschichten des Wafers, wobei entsprechend der Laufkarte unterschiedliche Prozesse pro Schritt zum Einsatz kommen können. Um diese geeignet digitalisieren zu können, müssen daher Möglichkeiten zur manuellen Interaktion innerhalb des Workflows bestehen, um die gewünschten Prozesse während dessen Ausführung auswählen zu können. Dieser Aspekt ist auch deshalb von Bedeutung, da es sich bei den eigentlichen Prozessschritten überwiegend um experimentelle Abläufe handelt, die nicht vollständig mithilfe eines Workflows automatisierbar sind. Dementsprechend ist ein generischer, interaktiver und möglichst einfach zu verwendender Ansatz zur Definition und Ausführung von Workflows notwendig, welcher die zugrunde liegenden Prozesse strukturiert und vollständig dokumentiert.

6.3.2 Ergebnisse

Als technische Grundlage für die Umsetzung dieses Fallbeispiels dient das Workflow-Management-System KadiStudio [240]. Bei diesem handelt es sich um ein im Kontext von Kadi4Mat entstandenes, prinzipiell jedoch davon unabhängiges Framework zur Definition und interaktiven Ausführung von Workflows. Erstere erfolgt mithilfe eines grafischen Editors, welcher das Hinzufügen und Manipulieren unterschiedlicher Knoten ermöglicht, die zu einem gerichteten Graphen verbunden werden können. Jeder dieser Knoten stellt entweder ein einzelnes

Werkzeug oder einen konfigurierbaren Eingangsparameter eines oder mehrerer Werkzeuge dar, während die Verbindungen der Knoten untereinander deren logische Abhängigkeiten definieren. Bei den Werkzeugen kann es sich sowohl um vordefinierte Funktionalitäten bzw. Knoten handeln, z. B. zur Implementierung von bedingten Anweisungen oder Interaktionen, als auch um benutzerdefinierte Werkzeuge. Diese Aspekte, sowie die disziplinunabhängige und generische Funktionsweise, unterscheiden KadiStudio von anderen Workflow-Management-Systemen [240].

Zur Ausführung eines Workflows wird eine externe Komponente eingesetzt, der sogenannte Process Manager [241], der wiederum eine oder mehrere Process Engines [242] verwaltet. Letztere sind für die eigentliche Ausführung von Workflows auf Grundlage der in einer den Workflow repräsentierenden Datei gespeicherten Informationen zuständig. Beide Komponenten werden über eine CLI angesteuert, sodass diese prinzipiell austauschbar sind, solange die entsprechenden Schnittstellen kompatibel zueinander sind. Im einfachsten Fall laufen alle der genannten Komponenten und Werkzeuge auf einem einzigen, lokalen System. Eine verteilte Ausführung ist unter Verwendung einer vom Process Manager bereitgestellten REST-API prinzipiell jedoch ebenfalls möglich.

Ähnlich wie bei den erläuterten Ausführungskomponenten funktioniert die Verwendung benutzerdefinierter Werkzeuge ebenfalls über eine CLI. Um dies zu ermöglichen, muss KadiStudio neben dem Namen des Werkzeugs dessen mögliche Parametrisierung kennen, was ebenfalls für die Generierung eines entsprechenden Knotens innerhalb des Editors notwendig ist. Dies erfolgt mithilfe einer XML-basierten Beschreibung, die ein Werkzeug in der Lage sein muss, bei Aufruf über die CLI mit dem Parameter `--xmlhelp` auf der Kommandozeile ausgeben zu können. Im Unterschied zu dem gängigen Parameter `--help`, der üblicherweise menschenlesbare Hilfetexte erzeugt, kann dadurch eine maschinenoperable Beschreibung und Parametrisierung erzeugt werden. Eine beispielhafte und gekürzte Ausgabe unter Verwendung eines aus kadi-apy stammenden CLI-Kommandos zur Erstellung eines Records in Kadi4Mat ist in Abbildung 6.12 dargestellt. Das resultierende XML setzt sich aus der innerhalb des `program`-Wurzelelements

```
<?xml version="1.0" encoding="utf-8"?>
<program name="kadi-apy records create"
         description="Create a record.">
    <param name="instance"
           char="I"
           type="string"
           description="Name of a Kadi instance defined in
            ↪ the config file."/>
    <param name="identifier"
           char="i"
           type="string"
           required="true"
           description="Identifier of the record."/>
    <param name="title"
           char="t"
           type="string"
           description="Title of the record."/>
</program>
```

Abbildung 6.12: Auszug aus der XML-basierten Hilfe eines aus kadi-apy stammenden CLI-Kommandos zur Erstellung eines Records in Kadi4Mat.

definierten Beschreibung des Werkzeugs selbst zusammen, sowie einer beliebigen Anzahl an `param`-Kindelementen zur Spezifikation der einzelnen Parameter. Die Attribute der jeweiligen Elemente enthalten sowohl für einen Nutzer hilfreiche Informationen, wie z. B. textuelle Beschreibungen zur Verwendung innerhalb des Workflow-Editors, als auch für die Ausführung relevante Angaben. Abbildung 6.13 zeigt einen basierend auf dieser XML-Beschreibung generierten Knoten. Neben den drei anhand der XML-Beschreibung dynamisch generierten Parametern, bei denen es sich jeweils um als Text zu interpretierende Werte handelt, sind ebenfalls fest eingebaute Eingabe- und Ausgabeverbindungen vorhanden. Diese umfassen die logischen Abhängigkeiten der Knoten untereinander (*Dependencies* und *Dependents*), die Möglichkeit für verkettete Kommandos zur Ausführung von Werkzeugen in bestimmen Umgebungen (*env*) und die Standardein- und -ausgabe (*stdin* und *stdout*) des Werkzeugs.

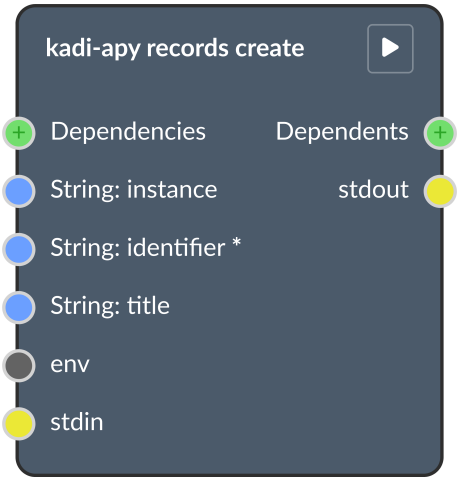

Abbildung 6.13: Anhand der XML-basierten Hilfe aus Abbildung 6.12 generierter Knoten zur Verwendung innerhalb eines Workflows.

Um in Bibliotheken wie kadi-apy eine passende Schnittstelle zur Generierung einer solchen XML-Beschreibung implementieren zu können, kann die Python-Bibliothek xmlhelpy [243] eingesetzt werden. Diese wurde speziell für solche Einsatzzwecke entwickelt und erweitert die bereits existierende Python-Bibliothek Click [244], die zur einfachen Implementierung und Parametrisierung von CLI-Werkzeugen verwendet werden kann. Neben der direkten Nutzung von xmlhelpy in Python-basierten Werkzeugen ist damit ebenfalls eine einfache Integration existierender CLI-Werkzeuge möglich, die selbst über keine `--xmlhelp`-Schnittstelle verfügen. In diesem Fall kann ein entsprechendes Werkzeug, das lediglich als Adapter dient, sämtliche Eingangsparameter entgegennehmen und diese anschließend an ein externes Werkzeug weiterleiten, z. B. in Form eines neuen Prozesses. Dadurch können ebenfalls die Parameter existierender Werkzeuge erweitert, eingeschränkt oder anderweitig modifiziert werden.

Zur Umsetzung des Fallbeispiels wurde, neben den von KadiStudio bereits mitgelieferten Knoten, lediglich von den durch kadi-apy bereitgestellten Werkzeugen

Gebrauch gemacht, wobei im Hintergrund erneut die zentral am KIT gehostete Instanz von Kadi4Mat zum Einsatz kam. In Bezug auf KadiStudio war aufgrund des variablen Ablaufs des hier betrachteten Workflows insbesondere die Verwendung interaktiver Funktionalitäten wichtig. Ein Beispiel einer entsprechenden Interaktion ist als Auszug des gesamten Workflows in Abbildung 6.14 dargestellt. Der

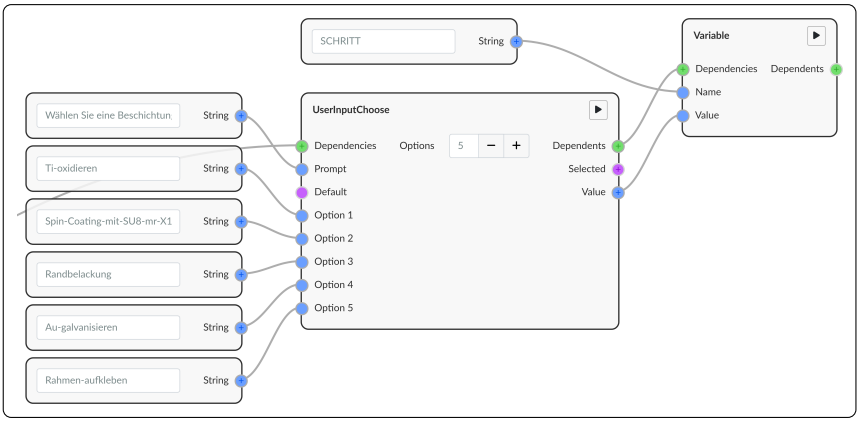

Abbildung 6.14: Auszug aus einem interaktiven Workflow zur Auswahl einer Beschichtungsmethode. Die unterschiedlichen Methoden sind als Parameter des interaktiven `UserInputChoose`-Knotens angegeben, wobei die vom Benutzer ausgewählte Methode bei Ausführung des Workflows in einer Variable mithilfe des `Variable`-Knotens zur späteren Verwendung zwischengespeichert wird.

abgebildete `UserInputChoose`-Knoten dient hierbei der interaktiven Auswahl einer von mehreren Beschichtungsmethoden des Wafers, wodurch ein entsprechender, variabler Schritt der zugehörigen Laufkarte repräsentiert wird. Neben den unterschiedlichen Methoden lässt sich ebenfalls eine Eingabeaufforderung (englisch: Prompt) parametrisieren, die bei Ausführung des Workflows zusätzlich zur Auswahl der Methoden dem Benutzer angezeigt wird. KadiStudio stellt hierzu ein geeignetes Formular bereit, um die Interaktion an passender Stelle entsprechend einer zuvor ermittelten Ausführungsreihenfolge anzuzeigen. Nach Auswahl der Beschichtungsmethode wird diese in Form einer Variable (`SCHRITT`) zwischengespeichert, was ebenfalls in Abbildung 6.14 in Form des `Variable`-Knotens

dargestellt ist. Variablen werden von der Process Engine verwaltet und können an unterschiedlichen Stellen im Workflow als Eingabeparameter mithilfe der Syntax `${VARIABLE}` wiederverwendet werden. So können sämtliche Schritte innerhalb der Laufkarte mithilfe ähnlicher Interaktionen repräsentiert und anschließend an zentraler Stelle unter Verwendung der `SCHRITT`-Variable abgearbeitet werden, wodurch redundante Abläufe innerhalb des Workflows vermieden werden.

Die Nutzung dieser Variable ist in Abbildung 6.15 dargestellt. Der abgebildete

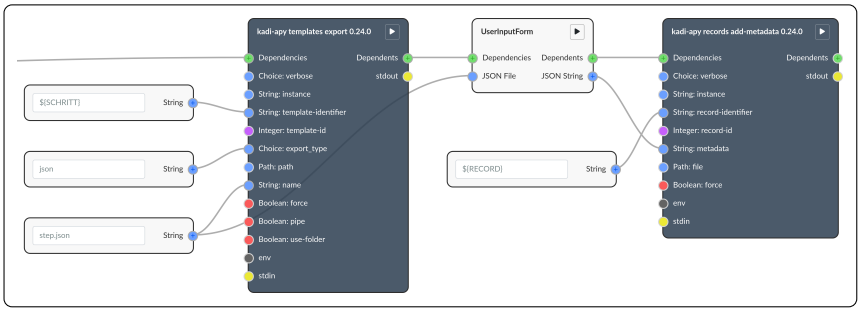

Abbildung 6.15: Auszug aus einem interaktiven Workflow zur Spezifikation generischer Record-Metadaten unter Verwendung eines Templates aus Kadi4Mat, wobei es sich insgesamt um denselben Workflow handelt, der teilweise bereits in Abbildung 6.14 dargestellt ist. Die Interaktion mit Kadi4Mat erfolgt mithilfe entsprechender, von kadi-apy bereitgestellter Werkzeuge bzw. Knoten, während die Metadaten unter Verwendung eines weiteren, interaktiven Knotens erfasst werden.

Auszug stellt den finalen Ablauf innerhalb des gesamten Workflows dar, welcher bei dessen Ausführung für den jeweils ausgewählten Prozessschritt erfolgt, wobei es sich in diesem Beispiel um das Beschichten des Wafers handelt. Der Ablauf besteht aus zwei von kadi-apy bereitgestellten sowie einem weiteren, interaktiven Knoten. Im ersten Schritt (`kadi-apy templates export`) wird unter Nutzung der `SCHRITT`-Variable ein Metadaten-Template von Kadi4Mat exportiert, das zuvor erstellt wurde und die Struktur der zu erfassenden, generischen Metadaten des entsprechenden, durch die Variable repräsentierten Prozesses enthält. Das exportierte Template wird dazu als lokale Datei auf dem zur Ausführung des Workflows verwendeten System im JSON-Format gespeichert und anschließend mithilfe des

`UserInputForm`-Knotens eingelesen, um in Form einer Interaktion dem Benutzer präsentiert zu werden. Dadurch wird eine einfache und grafische Spezifikation von Prozessparametern ähnlich der webbasierten GUI von Kadi4Mat ermöglicht. Unter Verwendung des letzten in Abbildung 6.15 dargestellten Knotens (`kadi-apy records add-metadata`) werden die resultierenden Metadaten als Teil eines existierenden Records hinterlegt, wobei hier ebenfalls eine Variable (`RECORD`) zur Angabe des Records in Form eines Identifiers zum Einsatz kommt. Der Record selbst lässt sich zu Beginn des Workflows ebenfalls durch eine Interaktion auswählen bzw. bei Durchführung des ersten Prozessschritts mithilfe eines entsprechenden Werkzeugs erstellen und repräsentiert nach Abarbeitung sämtlicher Schritte den vollständigen Fertigungsprozess des Wafers bzw. eine konkrete Ausprägung der in diesem Fallbeispiel betrachteten, variablen Laufkarte.

6.3.3 Fazit

Mithilfe des gezeigten Workflows konnte der Ablauf einer noch zu standardisierenden, variablen Laufkarte zur Herstellung von Siliziumwafern in digitaler Form aufgezeichnet werden. Auch in diesem Fallbeispiel lag der Schwerpunkt weniger auf der FAIRness erzeugter Metadaten, sondern auf der strukturierten und möglichst automatisierten Dokumentation der durchgeführten Prozesse und damit auch deren Reproduzierbarkeit, um eine zukünftige Standardisierung ermöglichen zu können. Zwar konnte bisher lediglich eine Teilautomatisierung erreicht werden, aufgrund des experimentellen Charakters dieses Fallbeispiels ist eine vollständige Automatisierung jedoch generell nur schwer möglich, weshalb insbesondere die Interaktionen innerhalb des gezeigten Workflows wichtig waren. Kadi4Mat nimmt in diesem Fallbeispiel primär die Rolle eines zentralen Datenspeichers ein, auf den mithilfe der Bibliothek kadi-apy lesend und schreibend zugegriffen wird. Durch die Spezifikation sämtlicher Metadaten eines Fertigungsprozesses innerhalb individueller Records kann ein digitaler Zwilling einer entsprechenden physischen Laufkarte erzeugt werden, was eine durchsuchbare Dokumentation der Arbeitsschritte ermöglicht. Während die Prozesse und die damit verbundene Datenherkunft prinzipiell ebenfalls unter Verwendung mehrerer Records sowie

passender Record-Links spezifiziert werden könnte, wird hierdurch eine rein Metadaten-basierte und dadurch simple Lösung realisiert.

Mithilfe der durch KadiStudio bereitgestellten Workflow-Funktionalität können prinzipiell beliebige Arbeitsabläufe modelliert werden. Das den Workflows zugrunde liegende EVA-Prinzip (Eingabe, Verarbeitung, Ausgabe), das jedem Knoten eine klar definierte Aufgabe zuweist, dient der einfachen Verständlichkeit der Workflows und erleichtert das Hinzufügen anwendungsspezifischer Werkzeuge. Neben kadi-apy werden durch Werkzeugsammlungen wie z. B. workflow-nodes [245] zusätzliche Funktionalitäten in Form einfach zu installierender Python-Pakete zur Verwendung innerhalb von Workflows bereitgestellt. Weiterhin befindet sich ein webbasierter Workflow-Editor als Teil der GUI von Kadi4Mat in Entwicklung, der alternativ ohne die Installation zusätzlicher Software zur Definition von Workflows genutzt werden kann und ebenfalls für die in diesem Abschnitt gezeigten Abbildungen verwendet wurde. Dieser stellt in Kombination mit KadiStudio und den im Rahmen dieses Fallbeispiels vorgestellten Werkzeugen eine weitere, wenn auch langfristige Perspektive für die direktere Unterstützung und Ausführung von Workflows in Kadi4Mat dar.

7 Diskussion

In diesem Kapitel werden die als Teil von Kadi4Mat umgesetzten Ergebnisse anhand der durchgeführten Fallbeispiele sowie für diesen Anwendungsfall angepasste Metriken zur Evaluation der FAIRness von Forschungsdaten untersucht. Anschließend wird ein Vergleich mit existierenden Systemen durchgeführt und die langfristigen Herausforderungen bei der Entwicklung und Etablierung einer VFU diskutiert.

7.1 Erfahrungsgestützte Evaluation

Die Verwendung der durch Kadi4Mat bereitgestellten Funktionalitäten wurde im Laufe dieser Arbeit sowohl anhand eines beispielhaften Anwendungsfalls als auch durch konkrete Fallbeispiele aufgezeigt. Zwar stellt ersterer einen idealistischen und stark vereinfachten Arbeitsablauf dar, jedoch lassen sich zumindest die umgesetzten Fallbeispiele für eine erfahrungsgestützte und praxisorientierte Evaluation von Kadi4Mat heranziehen.

Beim ersten Fallbeispiel, welches die Erzeugung FAIRer Forschungsdaten in der experimentellen Tribologie zum Ziel hatte, lag die Standardisierung und semantische Interoperabilität der erzeugten Metadaten im Fokus. Während Kadi4Mat in dieser Hinsicht auf Bottom-up-Ansätze spezialisiert ist, um je nach Anforderungen die schrittweise FAIRness von Forschungsdaten und Metadaten zu unterstützen, wurden die Ergebnisse dieses Fallbeispiels primär durch Top-down-Entwicklungen erzielt. Die dafür entwickelten semantischen Modelle und Werkzeuge machen die notwendigen Vorarbeiten deutlich, welche für den angestrebten

Detaillierungsgrad der Dokumentation sowie die Reproduzierbarkeit tribologischer Experimente erforderlich sein können. Insbesondere wurden diese benötigt, um die TriboDataFAIR-Ontologie mit Konzepten darzustellen, die mit Kadi4Mat kompatibel sind, wie z. B. generische Metadaten und Record-Links, da eine direkte Abbildung von Ontologien im aktuellen Entwicklungsstand nicht möglich ist. Dennoch wird durch das Fallbeispiel deutlich, wie Kadi4Mat als Brücke zwischen stark anwendungsspezifischen Entwicklungen und einem publizierten Datenpaket, welches die FAIR-Prinzipien in einem vergleichsweise hohen Maße erfüllt, genutzt werden kann. Die dazu notwendige Flexibilität wird hauptsächlich durch Kadi4Mats HTTP-API sowie darauf aufbauender Werkzeuge wie kadi-apy bereitgestellt. Weiterhin können verschiedene zusätzliche Funktionalitäten, die seit der Durchführung des Fallbeispiels implementiert wurden, direkt in Kadi4Mat zur verbesserten FAIRness des Datenpakets beitragen. Dazu gehören die Spezifikation von standardisierten Termen in Form von IRIs als Teil der generischen Record-Metadaten, ein RDF-basierter Export von Metadaten sowie die Nutzung von RO-Crates. Weitere Funktionalitäten, wie z. B. die kollaborative Erstellung eines zur Beschreibung des Vorzeigeexperiments geeigneten Vokabulars und darauf aufbauende Arbeiten, eignen sich dagegen vorrangig für die Umsetzung in Form einer separaten Entwicklung sowie einer entsprechenden Integration in Kadi4Mat. Dies wird auch durch die beiden weiteren Fallbeispiele deutlich, bei denen die Standardisierung der Metadaten eine geringere Rolle einnahm, sondern der Schwerpunkt auf der projekt- bzw. gruppeninternen Kollaboration unter Verwendung der von Kadi4Mat bereitgestellten Zugriffsrechteverwaltung lag.

Dem zweiten Fallbeispiel, bei dem eine datengetriebene Prozessüberwachung als Teil eines Ringversuchs im Bioprinting umgesetzt wurde, lag ebenfalls ein zuvor definierter Arbeitsablauf und damit ein entsprechender Top-down-Ansatz zugrunde. Dieser wurde jedoch zu Beginn und auch während der Durchführung des Ringversuchs kontinuierlich angepasst, wodurch eine praxisorientierte Digitalisierung der Prozessschritte erzielt werden konnte. Dabei wurde insbesondere von der GUI von Kadi4Mat Gebrauch gemacht, um sowohl Anleitungen in Form von Templates als auch die relevanten Prozessparameter und Daten in passende Records einzupflegen, weshalb vor allem die einfache Nutzung von Kadi4Mat im

Vergleich zu traditionellen, papierbasierten Laborbüchern wichtig war. Während im ersten Fallbeispiel ein Großteil der Arbeitsschritte von Experten durchgeführt wurde, die bereits über Vorwissen zu Kadi4Mat und FDM im Allgemeinen verfügten, waren bei der Durchführung des Ringversuchs größtenteils Forscher ohne entsprechende Fachkenntnisse und unterschiedlichster Disziplinen beteiligt. Auch wenn eine gewisse Einarbeitungszeit notwendig war, konnte gezeigt werden, dass dieser Aspekt keine große Hürde zur produktiven Verwendung von Kadi4Mat darstellt. Durch ein zusätzlich entwickeltes Werkzeug im Rahmen der durchgeführten Bildanalyse konnte ebenfalls der stark anwendungsspezifische Datentransfer in benutzerfreundlicher Art und Weise unter Nutzung der HTTP-API von Kadi4Mat durchgeführt werden. Dennoch sind weitere Möglichkeiten zur Bereitstellung integrierter Hilfestellungen und Vorgaben in Kadi4Mat für solche Fallbeispiele besonders wichtig, um die Notwendigkeit zusätzlicher Anleitungen möglichst auf anwendungsspezifische Arbeitsabläufe zu reduzieren.

Im letzten Fallbeispiel wurde lediglich von der HTTP-API von Kadi4Mat Gebrauch gemacht, um die Erzeugung von Records als Teil semi-automatisierter und interaktiver Workflows zu ermöglichen. Als zugrunde liegender Arbeitsfluss wurde die Produktion von Mikrostrukturbauteilen aus Siliziumwafern betrachtet, wobei der Fokus des Fallbeispiels auf den allgemeinen Aspekten lag, die zur Realisierung des Workflows notwendig waren. Kadi4Mat stellt bei dessen Ablauf einen zentralen Speicher für die Ablage von Metadaten bereit, der mithilfe passender Werkzeuge wie kadi-apy angesprochen wird. Auch wenn innerhalb des Workflows selbst keine direkte Interaktion mit Kadi4Mat aus Perspektive der Forscher stattfindet, konnte hierdurch erneut die flexible Umsetzung anwendungsspezifischer Arbeitsabläufe und die Möglichkeiten zur Einbettung in ein Workflow-Management-System gezeigt werden. Der erläuterte Workflow ist prinzipiell um beliebige Prozesse erweiterbar und damit ebenfalls allgemein genug, um auf ähnliche Arbeitsabläufe übertragen werden zu können. Dies kann ebenfalls die Verwendung weiterer FDM-Software unter Voraussetzung entsprechender Schnittstellen beinhalten, was eine weitere Möglichkeit zur Integration existierender Lösungen mit Kadi4Mat darstellt.

Zusammengefasst konnten die in den jeweiligen Fallbeispielen relevanten Zielsetzungen mithilfe von Kadi4Mat und darauf aufbauender Werkzeuge erfolgreich umgesetzt werden. Die generischen Funktionalitäten von Kadi4Mat wurden dabei um anwendungsspezifische Erweiterungen ergänzt, wobei insbesondere die HTTP-API eine zentrale Rolle einnahm. Im Hinblick auf den typischen Forschungsdatenlebenszyklus, welcher eine der Grundlagen der konzeptuellen Entwicklung von Kadi4Mat darstellt, kamen mit Ausnahme der Nachnutzung bereits publizierter Forschungsdaten sämtliche Phasen in unterschiedlichen Ausprägungen zum Einsatz. Da bei den Fallbeispielen jeweils Anwendungsfälle und Forschungsdisziplinen im Vordergrund standen, bei denen bisher wenig bis gar keine etablierten Standards für strukturiertes FDM existieren, lassen sich die umgesetzten Ansätze jedoch lediglich in Isolation bewerten. Weiterhin ist eine Evaluation möglicher Optimierungen durch den Übergang von analogen bzw. manuellen zu digitalen und (semi-)automatisierten Arbeitsabläufen erst dann vollständig möglich, wenn die entsprechenden Lösungen in die alltäglichen Forschungsarbeiten der jeweiligen Arbeitsgruppen integriert wurden. Im Kontext der Fallbeispiele wird dieser Schritt bereits im Rahmen experimenteller, tribologischer Arbeitsabläufe verfolgt, aufbauend auf den Arbeiten des ersten Fallbeispiels, wobei u. a. die zuvor erläuterten Weiterentwicklungen in Kadi4Mat zum Einsatz kommen. Ein weiterer Datensatz, welcher dem im Fallbeispiel erzeugten FAIR-Datenpaket ähnelt, die FAIR-Prinzipien im Hinblick auf die Interoperabilität jedoch in einem höheren Maße erfüllt, wurde ebenfalls bereits im Repositorium Zenodo veröffentlicht [246].

7.2 Qualitative Evaluation

Um die Funktionalitäten von Kadi4Mat unabhängig von den umgesetzten Fallbeispielen in allgemeiner Form qualitativ evaluieren zu können, bieten sich die FAIR-Prinzipien als eine mögliche Grundlage an. Zwar liegt deren Fokus auf der Publizierung und Nachnutzung von Forschungsdaten, die zur Erfüllung der Prinzipien notwendigen Voraussetzungen sind jedoch ebenfalls für allgemeingültige Aspekte wie z. B. der Qualität von Daten und Metadaten relevant. Zur

Evaluation der Prinzipien existiert eine Vielzahl unterschiedlicher Metriken, die in Form von Fragebögen, Checklisten oder vollautomatisiert mithilfe entsprechender Werkzeuge angewendet werden können [247]. Im ersten Fallbeispiel dieser Arbeit wurde bereits von den FAIRsFAIR Data Object Assessment Metrics Gebrauch gemacht, um die FAIRness des erzeugten Datenpakets evaluieren zu können. Diese bieten einen vergleichsweise hohen Detaillierungsgrad für die Bewertung der verschiedenen FAIR-Prinzipien und wurden bereits mehrfach auf der Grundlage entsprechenden Feedbacks überarbeitet, wobei die aktuelle Version zusätzliche Konformitätsstufen für die einzelnen Metriken definiert [37]. Da die Metriken in Form eines einfachen Dokuments vorliegen, eignen sich diese prinzipiell auch für die exemplarische Evaluation potenzieller Datensätze, die mithilfe einer spezifischen FDM-Software erzeugt und verwaltet werden können. Dennoch liegt der Fokus der FAIRsFAIR Data Object Assessment Metrics auf konkreten und individuellen Datensätzen, die innerhalb etablierter Repositorien publiziert wurden. Bei reiner Betrachtung der von Kadi4Mat bereitgestellten Funktionalitäten kann daher lediglich ein konzeptueller Überblick darüber gegeben werden, wie deren Nutzung potenziell zur FAIRness von verwalteten und optional publizierten Daten und Metadaten beitragen kann. Im weiteren Verlauf dieses Abschnitts wird eine Zusammenfassung der entsprechenden Evaluation vorgestellt, wobei eine vollständige Auflistung der einzelnen Metriken im Anhang in Abschnitt A.1 zu finden ist. Während bei deren Evaluation der Schwerpunkt auf Kadi4Mat selbst liegt, wird ebenfalls die Integration existierender Systeme berücksichtigt, was in diesem Fall insbesondere die Publizierung von Forschungsdaten mithilfe des Repositoriums Zenodo umfasst.

Die FAIRsFAIR Data Object Assessment Metrics lassen sich entsprechend der FAIR-Prinzipien in die Evaluation der Auffindbarkeit, Zugänglichkeit, Interoperabilität und Wiederverwendbarkeit publizierter Forschungsdaten aufteilen. Für Kadi4Mat selbst spielt insbesondere der Aspekt der Interoperabilität eine Rolle, da diese unabhängig davon wichtig ist, ob ein Datensatz publiziert, archiviert oder lediglich zur internen Kollaboration bereitgestellt wird, sowie die eng damit verwandte Wiederverwendbarkeit. Dabei liegt hauptsächlich die strukturierte Verwaltung von Metadaten im Fokus. Durch die von Kadi4Mat bereitgestellten

Funktionalitäten zur Spezifikation und Suche standardisierter Terme wird die Verwendung semantischer Ressourcen innerhalb der generischen Record-Metadaten ermöglicht. Diese können zusammen mit entsprechend abgebildeten Metadaten des Basisschemas von Kadi4Mat als Teil verschiedener Exportformate genutzt werden, wobei mithilfe des RDF-basierten Exports im Serialisierungsformat Turtle ebenfalls eine formale Wissensrepräsentationssprache zum Einsatz kommt. Bereits diese Möglichkeiten bieten ein vergleichsweise hohes Maß an Interoperabilität, erfordern jedoch das Vorhandensein eines geeigneten Vokabulars zur Beschreibung der jeweiligen Records. Deren Verlinkungen untereinander unterstützen wiederum die Angabe der Datenherkunft, zumindest für sämtliche, innerhalb von Kadi4Mat verwalteten Entitäten und Prozesse, und damit die Wiederverwendbarkeit der zugrunde liegenden Forschungsdaten, wobei hier ebenfalls die Nutzung semantischer Ressourcen möglich ist. Ein weiterer, für die Wiederverwendbarkeit relevanter Aspekt, ist die Nutzung etablierter Metadatenschemata und Datenformate entsprechend der Forschungsgemeinschaft, die in Zusammenhang mit den jeweiligen Daten und Metadaten steht. Ähnlich wie bei der potenziellen Nutzung standardisierter Terme, hängt dieser Aspekt maßgeblich von der jeweiligen Forschungsdisziplin ab und lässt sich daher nicht in allgemeiner Form evaluieren. Während Kadi4Mat in dieser Hinsicht absichtlich keine Vorgaben macht und generisch konzipiert ist, wird durch Funktionalitäten wie die Spezifikation generischer Metadaten und entsprechender Templates sowohl die Nutzung etablierter als auch die kollaborative Bottom-up-Entwicklung potenziell neuer Metadatenschemata ermöglicht. Weiterhin bietet der Export von Records und Collections in Form von RO-Crates ein einheitliches und standardisiertes Containerformat an, das unabhängig von den Formaten etwaiger Forschungsdaten ist.

Bei der Auffindbarkeit und Zugänglichkeit liegt dagegen der Fokus auf der möglichst öffentlich zugänglichen Publizierung von Forschungsdaten, was sich ebenfalls in den FAIRsFAIR Data Object Assessment Metrics widerspiegelt. Da Kadi4Mat hinsichtlich dieser Prinzipien von etablierten Repositorien wie Zenodo Gebrauch macht, sind die entsprechenden Metriken nur teilweise auf Kadi4Mat selbst anwendbar. Dies umfasst insbesondere die Vergabe von PIDs, die langfristig

gesicherte Zugänglichkeit von Metadaten und die Spezifikation externer Zugriffsbedingungen wie z. B. Embargofristen. Unabhängig davon sind für Kadi4Mat selbst dennoch Aspekte wie die maschinelle Lesbarkeit und Zugänglichkeit von Metadaten wichtig, die aufgrund der implementierten HTTP-API über ein standardisiertes Kommunikationsprotokoll und unter Nutzung der bereits erläuterten, maschinenoperablen Exportformate gegeben sind. Zusätzlich ist ein Großteil der von den Metriken definierten Kernelementen zur Auffindbarkeit von Metadaten, z. B. der Ersteller, Titel oder Identifikator eines Datensatzes, in vergleichbarer Form in Kadi4Mat als Teil des Basisschemas vorhanden. Diese können ebenfalls bei der Publizierung in Repositorien wie Zenodo auf die jeweiligen Schemata abgebildet werden.

7.3 Vergleich mit existierenden Systemen

Da der Fokus dieser Arbeit auf der Entwicklung von FDM-Software liegt, bietet sich neben der Evaluation von Kadi4Mat selbst auch ein abschließender Vergleich mit bereits existierenden Systemen bzw. mögliche Anknüpfungspunkte an diese an. Wie in der Gesamtübersicht des konzipierten Systems in Kapitel 4 erläutert, lässt sich Kadi4Mat logisch in zwei Komponenten aufteilen, die sich primär an den Funktionalitäten von ELNs und Repositorien orientieren. Daher bietet sich ein Vergleich insbesondere für diese Art von Systemen an. Während die ELN-Komponente durchaus klassische Funktionalitäten wie die Erfassung von Freitextbeschreibungen oder Skizzen umfasst, liegt deren Schwerpunkt im Vergleich zu den existierenden, in Kapitel 3 vorgestellten ELNs, auf der automatisierten Datenakquise unter Nutzung der bereitgestellten HTTP-API. Diese spielte ebenfalls innerhalb der umgesetzten Fallbeispiele eine zentrale Rolle. In Kombination mit der Repositorium-Komponente wird zusätzlich die strukturierte Verwaltung und der Austausch der akquirierten Forschungsdaten ermöglicht. Im Gegensatz zu existierenden Repositorien ist zwar keine direkte Publizierung von Forschungsdaten möglich, diese wird jedoch durch die Integration entsprechender

Systeme ermöglicht. Der Fokus von Kadi4Mat liegt somit auf der für die Publizierung notwendigen Aufbereitung verwalteter Daten und Metadaten, die ebenfalls für den internen Austausch oder die Archivierung relevant sein kann. Sonstige Arten von FDM-Systemen, wie DMP-Software, Workflow-Management-Systeme oder Registries, können in unterschiedlichen Ausprägungen unter Nutzung der umgesetzten HTTP-API und Pluginschnittstelle in Kadi4Mat integriert werden, wobei einige Möglichkeiten sowie konkrete Beispiele bereits im Verlauf dieser Arbeit erläutert wurden.

Im Hinblick auf den typischen Forschungsdatenlebenszyklus lässt sich Kadi4Mat damit primär in die Phasen der Erhebung und Aufbereitung von Forschungsdaten einordnen, teilweise aber auch in die Phasen der Publizierung, Archivierung und Nachnutzung. Die vollständige Abdeckung des gesamten Zyklus kann jedoch prinzipiell durch die erwähnte Integration existierender Systeme erreicht werden. Dadurch kann Kadi4Mat ebenfalls als Framework betrachtet werden, was einen maßgeblichen Unterschied zu bestehenden Systemen darstellt. Der Begriff Framework wurde bereits mehrfach im Laufe dieser Arbeit erwähnt und findet vor allem in der Softwaretechnik Anwendung zur Bezeichnung generischer Funktionalitäten, Bibliotheken oder Schnittstellen, die als Grundgerüst zur Entwicklung anwendungsspezifischer Applikationen dienen können. Im Kontext von Kadi4Mat bieten die bereitgestellten Schnittstellen ebenfalls verschiedene Anknüpfungspunkte, um als Framework zur Entwicklung benutzerdefinierter FDM-Lösungen eingesetzt werden zu können. Eine entsprechende Übersicht dieses Ansatzes ist in Abbildung 7.1 dargestellt. Schematisch lässt sich Kadi4Mat als webbasierte VFU in die drei Ebenen Frontend, Middleware und Backend aufteilen, wobei diese Einteilung lediglich eine mögliche Sichtweise darstellt. Das Frontend ist typischerweise nah am Endanwender angesiedelt und umfasst die von Kadi4Mat bereitgestellte GUI, die unter Verwendung entsprechender Applikationslogik als Teil der Middleware indirekt mit der technischen Infrastruktur des Backends interagiert. Die zuvor erwähnten Schnittstellen, welche für die Betrachtung von Kadi4Mat als Framework relevant sind, umfassen hauptsächlich die HTTP-API sowie die Möglichkeit zur Integration anwendungsspezifischer Plugins. Im Allgemeinen kann ebenfalls die

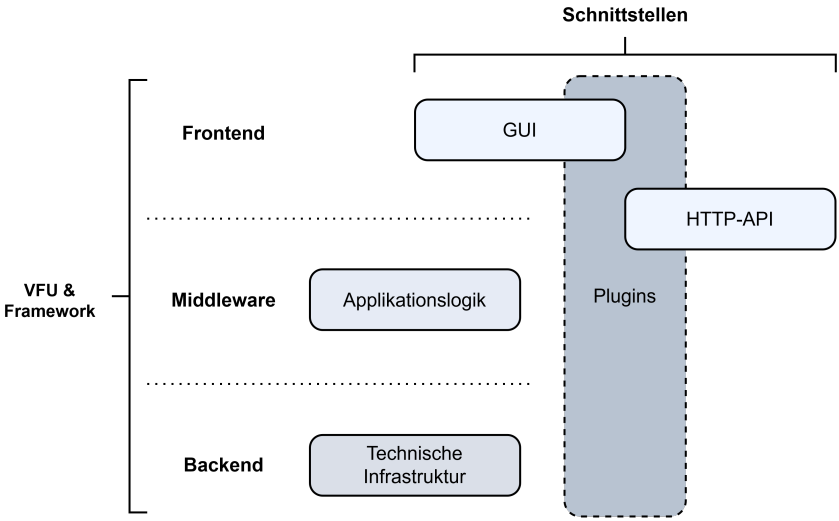

Abbildung 7.1: Schematische Darstellung von Kadi4Mat als Framework zur Entwicklung benutzerdefinierter FDM-Lösungen unter Nutzung verschiedener Schnittstellen.

GUI als benutzerorientierte, aber im Vergleich weniger flexible Schnittstelle betrachtet werden. Während sich die HTTP-API zwischen Frontend und Middleware einordnen lässt, können Plugins prinzipiell innerhalb sämtlicher Ebenen zum Einsatz kommen. Dies umfasst auch das Frontend, wodurch z. B. benutzerdefinierte Ansichten und Arbeitsflüsse als Teil der GUI von Kadi4Mat ermöglicht werden.

7.4 Langfristige Herausforderungen

Bereits die Erwägung, eine bestimmte FDM-Software lediglich innerhalb einer individuellen Arbeitsgruppe zu etablieren, kann durch verschiedene Faktoren beeinflusst werden. Diese sind ebenfalls für VFUs wie Kadi4Mat relevant, insbesondere da deren Komplexität in Bezug auf die Handhabung und Administration vergleichsweise hoch sein kann, während potenzielle Vorteile aus der Nutzung unter Umständen erst langfristig spürbar werden [165]. Neben den eigentlichen

Funktionalitäten, die eine VFU in einem für die jeweilige Forschungsdisziplin angemessenen und zuverlässigen Umfang bereitstellen muss, sollten diese ebenfalls möglichst einfach zu verwenden und in existierende Arbeitsabläufe integrierbar sein. Während letzteres durch die generische Entwicklung von Kadi4Mat und dazugehörige Schnittstellen ermöglicht wird, können genau diese Aspekte gleichzeitig die Benutzerfreundlichkeit einschränken. In den Fallbeispielen konnte bereits gezeigt werden, wie speziell entwickelte Werkzeuge als Teil benutzerspezifischer Anwendungsfälle zum Einsatz kommen können. Diese erfordern jedoch zusätzlichen Entwicklungsaufwand, weshalb Kadi4Mat auch ohne die Verwendung als Framework ein möglichst intuitives FDM ermöglichen sollte. Durch die Bereitstellung textueller Hilfestellungen sowie eines interaktiven Tutorials zur Erläuterung der grundlegenden Funktionalitäten bietet Kadi4Mat bereits verschiedene, integrierte Unterstützungsmöglichkeiten an, die zudem in mehreren Sprachen zur Verfügung gestellt werden. Die flexible Funktionsweise von Templates ermöglicht weiterhin unterschiedliche Optionen, um die manuelle Erfassung von Metadaten hinsichtlich der Elemente und Wertebereiche einzuschränken und kann zukünftig weiter in diese Richtung ausgebaut werden. Generell stellt der Mittelweg zwischen generischen und spezialisierten Lösungen einen wichtigen Aspekt dar, um die heterogenen Bedarfe der Ingenieurwissenschaften abzudecken und gleichzeitig benutzerdefinierte Arbeitsabläufe auch bereits ohne die Notwendigkeit umfangreicher, zusätzlicher Werkzeuge ermöglichen zu können.

Ein weiterer, wichtiger Faktor bei der Einführung einer VFU stellt deren Nachhaltigkeit dar [165]. Diese bezieht sich insbesondere auf die kontinuierliche Weiterentwicklung und Wartung des Codes, die speziell im wissenschaftlichen Umfeld an zeitlich begrenzte Forschungsprojekte bzw. Förderprogramme gebunden sein kann. Bei Kadi4Mat handelt es sich um eine quelloffene Software, die unter der Apache-Lizenz 2.0 veröffentlicht ist [167]. Die Entwicklung der Software findet unter Verwendung von GitLab [248] statt, ein webbasiertes System zur Versionsverwaltung von Softwareprojekten auf Basis des Werkzeugs Git [249]. Dadurch wird nicht nur eine kollaborative Entwicklung ermöglicht, sondern ebenfalls die Nachverfolgung und Erfassung von Problemen, Ideen für neue Funktionalitäten oder allgemeine Diskussionen, wozu keine Programmierkenntnisse erforderlich

sind. Dies fördert nicht nur den langfristigen Aufbau einer Nutzergemeinschaft auf Forscherebene, sondern ebenfalls die Beteiligung zusätzlicher Entwickler.

Zuletzt ist ebenfalls eine angemessene Administration von VFUs erforderlich, um eine stabile und sichere Nutzung zu gewährleisten. Dies bezieht sich nicht nur auf deren Erstinstallation, sondern auch auf die laufende Wartung, die z. B. Backups oder die kontinuierliche Aktualisierung notwendiger Softwarepakete umfassen kann. Wie bei den meisten VFUs ist aufgrund der webbasierten Umsetzung von Kadi4Mat eine zentralisierte Administration auf unterschiedlichen Ebenen möglich, z. B. auch innerhalb einzelner Arbeitsgruppen, was jedoch aufgrund begrenzter Hardware- und personeller Ressourcen einen erheblichen Mehraufwand darstellen kann. Dieser Aspekt war ebenfalls bei den in dieser Arbeit durchgeführten Fallbeispielen relevant, bei denen eine bereits verfügbare, sofort einsatzbereite Installation von Kadi4Mat zum Einsatz kam. Um diese Anwendungsfälle dennoch zu unterstützen, werden unterschiedliche Skripte und Installations- bzw. Administrationsanleitungen als Teil von Kadi4Mats Quellcode zur Verfügung gestellt [167]. Generell kann die Art der Installation für jeden Anwendungsfall stark variieren und von den Anforderungen an die Anpassungsfähigkeit, die Datenhoheit, die Möglichkeit zur Kollaboration oder bestimmten Sicherheitsbestimmungen abhängen.

8 Fazit

In der vorliegenden Arbeit wurde ein Konzept zur Unterstützung eines praxisorientierten und strukturierten FDMs in den Ingenieurwissenschaften mithilfe entsprechender Software entworfen. Die programmiertechnische Umsetzung dieses Konzepts wurde anschließend in Form des Systems Kadi4Mat sowie darauf aufbauender Werkzeuge realisiert und konnte im Rahmen unterschiedlicher Fallbeispiele erfolgreich in der Praxis angewandt werden. Anhand der u. a. darauf basierenden Evaluation des Systems konnte zudem gezeigt werden, wie sich dieses in den typischen Forschungsdatenlebenszyklus eingliedert und die Realisierung der FAIR-Prinzipien unterstützt. Weiterhin bestehen unterschiedliche Anknüpfungspunkte zu FDM-Architekturen wie FDO oder LDP sowie Möglichkeiten zur Einbettung in FDM-Initiativen bzw. föderierte Forschungsdateninfrastrukturen.

Kadi4Mat lässt sich am besten als VFU beschreiben, die aufgrund der generischen Funktionalitäten und Schnittstellen zur Integration existierender FDM-Software ebenfalls als Framework zur Entwicklung benutzerdefinierter FDM-Lösungen eingesetzt werden kann. Die generische Entwicklung ist insbesondere durch die heterogene und interdisziplinäre Natur der Ingenieurwissenschaften begründet und wird durch eine neuartige Kombination unterschiedlicher Komponenten aus ELNs, Repositorien und anderen Arten von FDM-Software ermöglicht. Zwar ist es nicht möglich, alle Forschungsdisziplinen innerhalb der Ingenieurwissenschaften individuell zu betrachten, dennoch zeigen die umgesetzten Ergebnisse das Potenzial, um in einer breiten Spanne von Anwendungsfällen genutzt werden zu können. Zusammengefasst konnte daher die innerhalb der zentralen Forschungsfrage dieser Arbeit definierte Zielsetzung erreicht werden. Weiterhin ist langfristig ebenfalls

der Einsatz von Kadi4Mat in anderen Disziplinen als den Ingenieurwissenschaften denkbar.

8.1 Ausblick

Die in dieser Arbeit beschriebenen und umgesetzten Ergebnisse können in verschiedener Hinsicht ausgebaut werden. Zum einen betrifft dies die technische Weiterentwicklung von Kadi4Mat, wobei etliche Möglichkeiten bereits im Laufe der Arbeit erläutert wurden. Diese umfassen u. a. die Erweiterung der bereitgestellten Template- und generischen Metadaten-Funktionalitäten, inklusive entsprechender Import- und Exportmöglichkeiten im Kontext des Semantic Webs, den Ausbau integrierter Hilfestellungen sowie zusätzliche Konfigurationsmöglichkeiten auf Ebene individueller Nutzer oder Systemadministratoren, um benutzer- bzw. anwendungsspezifische Anpassungen einfach ermöglichen zu können. Auch unterschiedliche Aspekte des Konzepts, die in der bisherigen Implementierung nicht betrachtet wurden, können unter diese Möglichkeiten fallen. Ein Beispiel stellt die (semi-)automatische Generierung von Metadaten dar, da Forscher zwar umfangreiche Metadaten zur Wiederverwendbarkeit von Forschungsdaten bevorzugen, jedoch nicht unbedingt die notwendige Zeit aufbringen können oder wollen, welche für die Erstellung qualitativ hochwertiger Metadaten erforderlich ist [122, 134]. Hier könnten z. B. Ansätze aus der künstlichen Intelligenz die Extraktion von Metadaten aus unterschiedlichen Datenquellen unterstützen [250]. Da sich zukünftig weiterhin die Frage stellen wird, welche Funktionalitäten Kadi4Mat selbst bereitstellen soll und kann, und welche davon lediglich in Form existierender Systeme und Werkzeuge integriert werden sollen, ist insbesondere jedoch die Erweiterung der HTTP-API und Pluginschnittstellen wichtig. Neben zusätzlichen Möglichkeiten zur Automatisierung kann dies ebenfalls für die Einbettung von Kadi4Mat in Forschungsdateninfrastrukturen relevant sein, die überwiegend mit z. B. aus datenschutzrechtlichen Gründen sensiblen Daten operieren [251].

Trotz den erläuterten Anknüpfungspunkten zwischen Kadi4Mat und existierenden Systemen wird langfristig weiterhin eine Vielzahl unterschiedlicher FDM-Software zum Einsatz kommen. Dies kann durch etablierte Arbeitsabläufe begründet sein, die potenziell bereits den gesamten bzw. den jeweils relevanten Teil des Forschungsdatenlebenszyklus abdecken, durch institutionelle Richtlinien oder der Notwendigkeit spezialisierter FDM-Lösungen, die sich mit generischen Funktionalitäten und zusätzlicher Werkzeuge nicht ausreichend realisieren lassen. Um dennoch den Austausch zwischen verschiedenen Systemen zu unterstützen, spielt die Interoperabilität verwalteter Ressourcen eine wichtige Rolle, die in Kadi4Mat u. a. durch die Verwendung standardisierter Exportformate erzielt wird. Ein Beispiel stellen RO-Crates dar, die sich zur strukturierten Bündelung von Forschungsdaten eignen und die Spezifikation grundlegender Metadaten als Teil der im Archiv enthaltenen Metadatendatei ermöglichen. RO-Crates bilden ebenfalls die Grundlage bei der Etablierung eines systemübergreifenden Austauschformats für Forschungsdaten im Kontext des sogenannten ELN-Konsortiums [252], dem u. a. Kadi4Mat sowie die in Kapitel 3 vorgestellten ELNs angehören. Bei diesem handelt es sich um einen losen Zusammenschluss verschiedener ELNs und vergleichbarer Systeme mit dem Ziel, gemeinsame Spezifikationen zu erarbeiten, um langfristig ein grundlegendes Maß an plattformübergreifender Interoperabilität zu gewährleisten.

Wie bereits in Kapitel 4 erwähnt, können neben der Etablierung von FDM-Software auch andere Ansätze einen wichtigen Stellenwert bei der Umsetzung eines strukturierten FDMs einnehmen. Dies betrifft insbesondere die Thematik der Datenkompetenzen, die unabhängig vom Einsatz konkreter Software relevant sind und bereits einfache Praktiken wie die konsistente Benennung lokal verwalteter Dateien umfassen können. Für den Aufbau entsprechender Kompetenzen bieten sich z. B. Schulungen oder Workshops an [253], jedoch ist ebenfalls eine Integration in Lehrpläne möglich [75, 254]. Dieser Aspekt geht Hand in Hand mit der Entwicklung von FDM-Software, weshalb beide Ansätze verstärkt gemeinsam weiterentwickelt werden müssen.

Literaturverzeichnis

[1] T. Hey und A. Trefethen, „The Data Deluge: An e-Science Perspective“, in *Wiley Series in Communications Networking & Distributed Systems*, F. Berman, G. Fox und T. Hey, Hrsg., Chichester, UK: John Wiley & Sons, Ltd, 11. März 2003, S. 809–824. DOI: 10.1002/0470867167.ch36.

[2] C. L. Borgman, „The conundrum of sharing research data“, *Journal of the American Society for Information Science and Technology*, Jg. 63, Nr. 6, S. 1059–1078, Juni 2012. DOI: 10.1002/asi.22634.

[3] S. Sandfeld, T. Dahmen, F. O. R. Fischer, C. Eberl, S. Klein, M. Selzer, J. Möller, F. Mücklich, M. Engstler, S. Diebels, R. Tschuncky, A. Prakash, D. Steinberger, C. Kübel, H.-G. Herrmann und R. Schubotz, „Strategiepapier - Digitale Transformation in der Materialwissenschaft und Werkstofftechnik“, Deutsche Gesellschaft für Materialkunde e.V., Frankfurt, Germany, 2018.

[4] J. Kimmig, S. Zechel und U. S. Schubert, „Digital Transformation in Materials Science: A Paradigm Change in Material's Development“, *Advanced Materials*, Jg. 33, Nr. 8, S. 2 004 940, Feb. 2021. DOI: 10.1002/adma.202004940.

[5] A. J. G. Hey, Hrsg., *The fourth paradigm: data-intensive scientific discovery*. Redmond, Washington: Microsoft Research, 2009, 251 S., ISBN: 978-0-9825442-0-4.

[6] Deutsche Forschungsgemeinschaft, „Guidelines for Safeguarding Good Research Practice. Code of Conduct“, 20. Apr. 2022. DOI: 10.5281/ZENODO.6472827.

[7] European Parliament and Council, „Regulation (EU) 2021/695“, *Official Journal of the European Union*, 28. Apr. 2021.

[8] M. D. Wilkinson, M. Dumontier, IJ. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao und B. Mons, „The FAIR Guiding Principles for scientific data management and stewardship“, *Scientific Data*, Jg. 3, Nr. 1, Dez. 2016. DOI: 10.1038/sdata.2016.18.

[9] „NFDI4Ing“. (2023), Adresse: https://nfdi4ing.de (eingesehen am 10. 08. 2023).

[10] Deutsche Forschungsgemeinschaft, „Leitlinien zum Umgang mit Forschungsdaten“, Deutsche Forschungsgemeinschaft, 30. Sep. 2015.

[11] M. Kindling und P. Schirmbacher, „„Die digitale Forschungswelt“ als Gegenstand der Forschung / Research on Digital Research / Recherche dans la domaine de la recherche numérique“, *Information - Wissenschaft & Praxis*, Jg. 64, Nr. 2-3, 1. Apr. 2013. DOI: 10.1515/iwp-2013-0017.

[12] ISO Central Secretary, „Graphic technology — Extensible metadata platform (XMP) — Part 1: Data model, serialization and core properties“, International Organization for Standardization, Geneva, CH, Standard ISO 16684-1:2019, Apr. 2019.

[13] The HDF Group. „Hierarchical Data Format, version 5“. (1997), Adresse: https://www.hdfgroup.org/HDF5 (eingesehen am 10. 08. 2023).

[14] S. Soiland-Reyes, P. Sefton, M. Crosas, L. J. Castro, F. Coppens, J. M. Fernández, D. Garijo, B. Grüning, M. La Rosa, S. Leo, E. Ó Carragáin, M. Portier, A. Trisovic, RO-Crate Community, P. Groth und C. Goble, „Packaging research artefacts with RO-Crate“, *Data Science*, Jg. 5, Nr. 2, S. Peroni, Hrsg., S. 97–138, 20. Juli 2022. DOI: 10.3233/DS-210053.

[15] J. Kunze, J. Littman, E. Madden, J. Scancella und C. Adams, „The BagIt File Packaging Format (V1.0)“, RFC Editor, RFC 8493, Okt. 2018, S. 1–25.

[16] Research Data Alliance. „Metadata Standards Catalog“. (2023), Adresse: https://rdamsc.bath.ac.uk (eingesehen am 10. 08. 2023).

[17] M. Bauer, H. Baqa, S. Bilbao, A. Corchero, L. Daniele, I. Esnaola, I. Fernandez, O. Franberg, R. Garcia-Castro, M. Girod-Genet, P. Guillemin, A. Gyrard, C. E. Kaed, A. Kung, J. Lee, M. Lefrancois, W. Li, D. Raggett und M. Wetterwald, „Towards Semantic Interoperability Standards based on Ontologies“, 2019. DOI: 10.13140/RG.2.2.26825.29282.

[18] T. R. Gruber, „A translation approach to portable ontology specifications“, *Knowledge Acquisition*, Jg. 5, Nr. 2, S. 199–220, Juni 1993. DOI: 10.1006/knac.1993.1008.

[19] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator und W. Swartout, „Enabling technology for knowledge sharing.“, *AI Magazine*, Jg. 12, S. 36–56, Sep. 1991.

[20] T. Pellegrini und A. Blumauer, Hrsg., *Semantic Web: Wege zur vernetzten Wissensgesellschaft* (X.media.press). Berlin Heidelberg: Springer, 2006, 533 S., ISBN: 978-3-540-29324-8.

[21] T. Berners-Lee, J. Hendler und O. Lassila, „The semantic web“, *Scientific american*, Jg. 284, Nr. 5, S. 34–43, 2001.

[22] S. Bechhofer, F. Van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider und L. A. Stein, „OWL Web Ontology Language Reference“, World Wide Web Consortium, Cambridge, MA, USA, W3C Recommendation, 2004.

[23] „Resource Description Framework (RDF): Concepts and Abstract Syntax“, World Wide Web Consortium, Cambridge, MA, USA, W3C Recommendation, 2004.

[24] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, P.-A. Champin und N. Lindström, „JSON-LD 1.1“, World Wide Web Consortium, Cambridge, MA, USA, W3C Recommendation, 2020.

[25] D. Beckett, T. Berners-Lee, E. Prud'hommeaux und G. Carothers, „RDF 1.1 Turtle“, World Wide Web Consortium, Cambridge, MA, USA, W3C Recommendation, 2014.

[26] S. Büttner, H.-C. Hobohm und L. Müller, Hrsg., *Handbuch Forschungsdatenmanagement*. Bad Honnef: Bock + Herchen Verlag, 2011, 223 S., ISBN: 978-3-88347-283-6.

[27] A. Surkis und K. Read, „Research data management“, *Journal of the Medical Library Association*, Jg. 103, Nr. 3, S. 154–156, Juli 2015. DOI: 10.3163/1536-5050.103.3.011.

[28] M. Putnings, H. Neuroth und J. Neumann, Hrsg., *Praxishandbuch Forschungsdatenmanagement*. De Gruyter, 18. Jan. 2021. DOI: 10.1515/9783110657807.

[29] S. Higgins, „The DCC Curation Lifecycle Model“, *International Journal of Digital Curation*, Jg. 3, Nr. 1, S. 134–140, 2. Dez. 2008. DOI: 10.2218/ijdc.v3i1.48.

[30] A. Ball, „Review of Data Management Lifecycle Models“, University of Bath, Bath, UK, Project Document, 10. Jan. 2012.

[31] C. Strasser, *Research Data Management*. 2015, ISBN: 978-1-937522-65-0.

[32] E. White, E. Baldridge, Z. Brym, K. Locey, D. McGlinn und S. Supp, „Nine simple ways to make it easier to (re)use your data“, *Ideas in Ecology and Evolution*, Jg. 6, Nr. 2, 2013. DOI: 10.4033/iee.2013.6b.6.f.

[33] G. K. Sandve, A. Nekrutenko, J. Taylor und E. Hovig, „Ten Simple Rules for Reproducible Computational Research“, *PLoS Computational Biology*, Jg. 9, Nr. 10, P. E. Bourne, Hrsg., e1003285, 24. Okt. 2013. DOI: 10.1371/journal.pcbi.1003285.

[34] B. Mons, C. Neylon, J. Velterop, M. Dumontier, L. O. B. da Silva Santos und M. D. Wilkinson, „Cloudy, increasingly FAIR; revisiting the FAIR Data guiding principles for the European Open Science Cloud“, *Information Services & Use*, Jg. 37, Nr. 1, S. 49–56, 7. März 2017. DOI: 10.3233/ISU-170824.

[35] M. D. Wilkinson, R. Verborgh, L. O. Bonino da Silva Santos, T. Clark, M. A. Swertz, F. D. Kelpin, A. J. Gray, E. A. Schultes, E. M. van Mulligen, P. Ciccarese, A. Kuzniar, A. Gavai, M. Thompson, R. Kaliyaperumal, J. T. Bolleman und M. Dumontier, „Interoperability and FAIRness through a novel combination of Web technologies“, *PeerJ Computer Science*, Jg. 3, Nr. e110, 24. Apr. 2017. DOI: 10.7717/peerj-cs.110.

[36] A.-L. Lamprecht, L. Garcia, M. Kuzak, C. Martinez, R. Arcila, E. Martin Del Pico, V. Dominguez Del Angel, S. van de Sandt, J. Ison, P. A. Martinez, P. McQuilton, A. Valencia, J. Harrow, F. Psomopoulos, J. L. Gelpi, N. Chue Hong, C. Goble und S. Capella-Gutierrez, „Towards FAIR principles for research software“, *Data Science*, Jg. 3, Nr. 1, P. Groth und M. Dumontier, Hrsg., S. 37–59, 12. Juni 2020. DOI: 10.3233/DS-190026.

[37] A. Devaraju, R. Huber, M. Mokrane, P. Herterich, L. Cepinskas, J. Vries, H. L'Hours, J. Davidson und A. White, „FAIRsFAIR Data Object Assessment Metrics“, Version 0.5, 14. Apr. 2022. DOI: 10.5281/ZENODO.6461229.

[38] S. Hettrick, M. Antonioletti, L. Carr, N. Chue Hong, S. Crouch, D. De Roure, I. Emsley, C. Goble, A. Hay, D. Inupakutika, M. Jackson, A. Nenadic, T. Parkinson, M. I. Parsons, A. Pawlik, G. Peru, A. Proeme, J. Robinson und S. Sufi, *UK Research Software Survey 2014*, Zenodo, 4. Dez. 2014. DOI: 10.5281/ZENODO.14809.

[39] C. R. Prause, R. Reiners und S. Dencheva, „Empirical Study of Tool Support in Highly Distributed Research Projects“, in *2010 5th IEEE International Conference on Global Software Engineering*, Princeton, NJ, USA: IEEE, Aug. 2010, S. 23–32. DOI: 10.1109/ICGSE.2010.13.

[40] DCC, „Checklist for a Data Management Plan“, Digital Curation Centre, Edinburgh, v.4.0, 2013.

[41] W. K. Michener, „Ten Simple Rules for Creating a Good Data Management Plan“, *PLOS Computational Biology*, Jg. 11, Nr. 10, P. E. Bourne, Hrsg., e1004525, 22. Okt. 2015. DOI: 10.1371/journal.pcbi.1004525.

[42] A. Sallans und M. Donnelly, „DMP Online and DMPTool: Different Strategies Towards a Shared Goal“, *International Journal of Digital Curation*, Jg. 7, Nr. 2, S. 123–129, 23. Okt. 2012. DOI: 10.2218/ijdc.v7i2.235.

[43] R. Pergl, R. Hooft, M. Suchánek, V. Knaisl und J. Slifka, „“Data Stewardship Wizard”: A Tool Bringing Together Researchers, Data Stewards, and Data Experts around Data Management Planning“, *Data Science Journal*, Jg. 18, S. 59, 19. Dez. 2019. DOI: 10.5334/dsj-2019-059.

[44] J. Klar, O. Michaelis, C. Engelhardt, H. Enke, J. Frenzel, D. Hausen, G. Jagusch, C. Kramer, B. Lindstädt, J. Ludwig, N. Heike, J. Straka, R. Strötgen, R. Ulrich, K. Wedlich-Zachodin, U. Wuttke, G. Lanza, D. Martínez Muñoz und D. Pilori, *Research Data Management Organizer (RDMO)*, Version 1.11.0, Zenodo, 1. Aug. 2023. DOI: 10.5281/ZENODO.8206604.

[45] H. K. Machina und D. J. Wild, „Electronic Laboratory Notebooks Progress and Challenges in Implementation“, *SLAS Technology*, Jg. 18, Nr. 4, S. 264–268, Aug. 2013. DOI: 10.1177/2211068213484471.

[46] E. D. Foster, E. C. Whipple und G. R. Rios, „Implementing an institution-wide electronic lab notebook initiative“, *Journal of the Medical Library Association*, Jg. 110, Nr. 2, 26. Apr. 2022. DOI: 10.5195/jmla.2022.1407.

[47] M. Rubacha, A. K. Rattan und S. C. Hosselet, „A Review of Electronic Laboratory Notebooks Available in the Market Today“, *Journal of Laboratory Automation*, Jg. 16, Nr. 1, S. 90–98, Feb. 2011. DOI: 10.1016/j.jala.2009.01.002.

[48] Harvard Longwood Medical Area Research Data Management Working Group, „Electronic Lab Notebook Comparison Matrix“, 19. Mai 2021. DOI: 10.5281/ZENODO.4723753.

[49] C. L. Bird, C. Willoughby und J. G. Frey, „Laboratory notebooks in the digital era: The role of ELNs in record keeping for chemistry and other sciences“, *Chemical Society Reviews*, Jg. 42, Nr. 20, S. 8157, 2013. DOI: 10.1039/c3cs60122f.

[50] K. T. Taylor, „The status of electronic laboratory notebooks for chemistry and biology“, *Current Opinion in Drug Discovery and Development*, Jg. 9, Nr. 3, S. 348–353, 2006.

[51] J. Potthoff, P. Tremouilhac, P. Hodapp, B. Neumair, S. Bräse und N. Jung, „Procedures for systematic capture and management of analytical data in academia“, *Analytica Chimica Acta: X*, Jg. 1, S. 100 007, März 2019. DOI: 10.1016/j.acax.2019.100007.

[52] C. A. Lynch, „Institutional Repositories: Essential Infrastructure For Scholarship In The Digital Age“, *portal: Libraries and the Academy*, Jg. 3, Nr. 2, S. 327–336, 2003. DOI: 10.1353/pla.2003.0039.

[53] P. B. Heidorn, „Shedding Light on the Dark Data in the Long Tail of Science“, *Library Trends*, Jg. 57, Nr. 2, S. 280–299, 2008. DOI: 10.1353/lib.0.0036.

[54] R. C. Amorim, J. A. Castro, J. Rocha da Silva und C. Ribeiro, „A comparison of research data management platforms: Architecture, flexible metadata and interoperability“, *Universal Access in the Information Society*, Jg. 16, Nr. 4, S. 851–862, Nov. 2017. DOI: 10.1007/s10209-016-0475-y.

[55] S. Stall, M. E. Martone, I. Chandramouliswaran, L. Federer, J. Gautier, J. Gibson, M. Hahnel, J. Larkin, N. Pfeiffer, B. Sedora, I. Sim, T. Smith, A. E. Van Gulick, E. Walker, J. Wood, M. Zaringhalam und A. Zigoni, „Generalist Repository Comparison Chart“, Version 3.0, 17. Mai 2023. DOI: 10.5281/ZENODO.7946938.

[56] D. Lin, J. Crabtree, I. Dillo, R. R. Downs, R. Edmunds, D. Giaretta, M. De Giusti, H. L'Hours, W. Hugo, R. Jenkyns, V. Khodiyar, M. E. Martone, M. Mokrane, V. Navale, J. Petters, B. Sierman, D. V. Sokolova, M. Stockhause und J. Westbrook, „The TRUST Principles for digital repositories“, *Scientific Data*, Jg. 7, Nr. 1, Dez. 2020. DOI: 10.1038/s41597-020-0486-7.

[57] I. Dillo und L. De Leeuw, „CoreTrustSeal“, *Mitteilungen der Vereinigung Österreichischer Bibliothekarinnen und Bibliothekare*, Jg. 71, Nr. 1, S. 162–170, 19. Juli 2018. DOI: 10.31263/voebm.v71i1.1981.

[58] S. Jablonski, „Architektur von Workflow-Management-Systemen“, *Informatik - Forschung und Entwicklung*, Jg. 12, Nr. 2, S. 72–81, 15. Mai 1997. DOI: 10.1007/s004500050076.

[59] S. Bowers, „Scientific Workflow, Provenance, and Data Modeling Challenges and Approaches“, *Journal on Data Semantics*, Jg. 1, Nr. 1, S. 19–30, Mai 2012. DOI: 10.1007/s13740-012-0004-y.

[60] Directorate-General for Research and Innovation, „Turning Fair Into Reality - Final Report and Action Plan from the European Commission Expert Group and FAIR Data“, 26. Nov. 2018.

[61] D. Tudhope, T. Koch und R. Heery, *Terminology services and technology: JISC state of the art review*. Joint Information Systems Committee, Sep. 2006.

[62] EOSC Future. „EOSC Portal“. (2023), Adresse: https://eosc-portal.eu (eingesehen am 10. 08. 2023).

[63] RfII - Rat für Informationsinfrastrukturen, „Föderierte Dateninfrastrukturen für die wissenschaftliche Nutzung. NFDI, EOSC und Gaia-X: Vergleich und Anregungen für eine engagierte Mitgestaltung des Ausbaus und der Weiterentwicklung“, Göttingen, 4, 2023, S. 48.

[64] Nationale Forschungsdateninfrastruktur e. V. „NFDI“. (2023), Adresse: https://www.nfdi.de (eingesehen am 10. 08. 2023).

[65] R. Schmitt, M. Müller, P. Pelz, T. Stäcker, T. Bronger, I. Sens und A. Streit, „NFDI4ING: Eine nationale Forschungsdateninfrastruktur für die Ingenieurwissenschaften“, Institutionelles Repositorium der Leibniz Universität Hannover, 2018.

[66] A. M. Cox, M. A. Kennan, L. Lyon und S. Pinfield, „Developments in research data management in academic libraries: Towards an understanding of research data service maturity“, *Journal of the Association for Information Science and Technology*, Jg. 68, Nr. 9, S. 2182–2200, Sep. 2017. DOI: 10.1002/asi.23781.

[67] M. Politze und T. Eifert, „On the Decentralization of IT Infrastructures for Research Data Management“, in *EUNIS 2019 Congress*, Trondheim: Norwegian University of Science and Technology (NTNU), 4. Juni 2019, S. 108–109.

[68] C. Curdt, J. Dierkes und S. Kloppenburg, „RDM in a Decentralised University Ecosystem—A Case Study of the University of Cologne“, *Data Science Journal*, Jg. 21, S. 20, 27. Dez. 2022. DOI: 10.5334/dsj-2022-020.

[69] „Baden-Württembergisches Begleit- und Weiterentwicklungsprojekt für Forschungsdatenmanagement (bw2FDM)“. (2023), Adresse: https://bwfdm.scc.kit.edu/index.php (eingesehen am 10. 08. 2023).

[70] A. M. Cox und S. Pinfield, „Research data management and libraries: Current activities and future priorities“, *Journal of Librarianship and Information Science*, Jg. 46, Nr. 4, S. 299–316, Dez. 2014. DOI: 10.1177/0961000613492542.

[71] A. Andrikopoulou, J. Rowley und G. Walton, „Research Data Management (RDM) and the Evolving Identity of Academic Libraries and Librarians: A Literature Review“, *New Review of Academic Librarianship*, Jg. 28, Nr. 4, S. 349–365, 2. Okt. 2022. DOI: 10.1080/13614533.2021.1964549.

[72] A. Cox, E. Verbaan und B. Sen, „Upskilling Liaison Librarians for Research Data Management“, *Ariadne*, Nr. 70, 2012, ISSN: 1361-3200.

[73] C. Wendelborn, M. Anger und C. Schickhardt, „What is data stewardship? Towards a comprehensive understanding“, *Journal of Biomedical Informatics*, Jg. 140, S. 104 337, Apr. 2023. DOI: 10.1016/j.jbi.2023.104337.

[74] „POLiS - Post Lithium Storage Cluster of Excellence“. (2023), Adresse: https://www.postlithiumstorage.org (eingesehen am 10. 08. 2023).

[75] Y. Demchenko und L. Stoy, „Research Data Management and Data Stewardship Competences in University Curriculum“, präsentiert auf 2021 IEEE Global Engineering Education Conference (EDUCON), Vienna, Austria: IEEE, 21. Apr. 2021, S. 1717–1726. DOI: 10.1109/EDUCON46332.2021.9453956.

[76] N. S. Redkina, „Current Trends in Research Data Management“, *Scientific and Technical Information Processing*, Jg. 46, Nr. 2, S. 53–58, Apr. 2019. DOI: 10.3103/S0147688219020035.

[77] B. Heinrichs, M. Politze und M. Yazdi, „Evaluation of Architectures for FAIR Data Management in a Research Data Management Use Case:“ In *Proceedings of the 11th International Conference on Data Science, Technology and Applications*, Lisbon, Portugal: SCITEPRESS - Science

and Technology Publications, 2022, S. 476–483. DOI: 10.5220/0011302700003269.

[78] E. Schultes und P. Wittenburg, „FAIR Principles and Digital Objects: Accelerating Convergence on a Data Infrastructure“, in *Data Analytics and Management in Data Intensive Domains*, Y. Manolopoulos und S. Stupnikov, Hrsg., Bd. 1003, Cham: Springer International Publishing, 2019, S. 3–16. DOI: 10.1007/978-3-030-23584-0_1.

[79] K. De Smedt, D. Koureas und P. Wittenburg, „FAIR Digital Objects for Science: From Data Pieces to Actionable Knowledge Units“, *Publications*, Jg. 8, Nr. 2, S. 21, 11. Apr. 2020. DOI: 10.3390/publications8020021.

[80] L. O. B. da Silva Santos. „FAIR Digital Object Framework Documentation“. unter Mitarb. von G. Guizzardi und T. Prince Sales. (2022), Adresse: https://fairdigitalobjectframework.org (eingesehen am 10. 08. 2023).

[81] J. Schweikert, K.-U. Stucky, W. Süß und V. Hagenmeyer, „A Photovoltaic System Model Integrating FAIR Digital Objects and Ontologies“, *Energies*, Jg. 16, Nr. 3, S. 1444, 1. Feb. 2023. DOI: 10.3390/en16031444.

[82] „Linked Data Platform 1.0“, World Wide Web Consortium, Cambridge, MA, USA, W3C Recommendation, 2015.

[83] ISO Central Secretary, „Information and documentation — The Dublin Core metadata element set — Part 1: Core elements“, International Organization for Standardization, Geneva, CH, Standard ISO 15836-1:2017, Mai 2017.

[84] DataCite Metadata Working Group, „DataCite Metadata Schema Documentation for the Publication and Citation of Research Data and Other Research Outputs v4.4“, Version 4.4, unter Mitarb. von M. de Smaele, I. Bernal Martínez, R. Dasler, Ashton, S. Roy, M. Fenner, L. Chiloane, M. Burger, M. Yahia, L. Zolly, T. Habermann, A. Raugh, V. Ilik und S. Foulger, 2021. DOI: 10.14454/3W3Z-SA82.

[85] DataCite Metadata Working Group, „DataCite to Dublin Core Mapping v4.4.“, Version 4.4, unter Mitarb. von M. de Smaele, B. Martínez, R. Dasler, J. Ashton, S. Roy, M. Fenner, L. Chiloane, M. Burger, M. Yahia, L. Zolly, T. Habermann, A. Raugh, V. Ilik und S. Foulger, 2021. DOI: 10.14454/QN00-QX85.

[86] B. Schembera und D. Iglezakis, „EngMeta: Metadata for Computational Engineering“, *International Journal of Metadata, Semantics and Ontologies*, Jg. 14, Nr. 1, S. 26, 2020. DOI: 10.1504/IJMSO.2020.107792.

[87] „PROV-Overview: An Overview of the PROV Family of Documents“, World Wide Web Consortium, Cambridge, MA, USA, W3C Recommendation, 2013.

[88] „Dublin Core to PROV Mapping“, World Wide Web Consortium, Cambridge, MA, USA, W3C Recommendation, 2013.

[89] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan und J. V. den Bussche, „The Open Provenance Model core specification (v1.1)“, *Future Generation Computer Systems*, Jg. 27, Nr. 6, S. 743–756, Juni 2011. DOI: 10.1016/j.future.2010.07.005.

[90] Google LLC, Microsoft Corporation, Yahoo!, Inc. und Yandex N.V. „Schema.org“. (2011), Adresse: https://schema.org (eingesehen am 10. 08. 2023).

[91] „RDF 1.2 Schema“, World Wide Web Consortium, Cambridge, MA, USA, W3C Recommendation, 2023.

[92] „Data Catalog Vocabulary (DCAT) - Version 2“, World Wide Web Consortium, Cambridge, MA, USA, W3C Recommendation, 2020.

[93] N. CARPi, A. Minges und M. Piel, „eLabFTW: An open source laboratory notebook for research labs“, *The Journal of Open Source Software*, Jg. 2, Nr. 12, S. 146, 14. Apr. 2017. DOI: 10.21105/joss.00146.

[94] P. Tremouilhac, A. Nguyen, Y.-C. Huang, S. Kotov, D. S. Lütjohann, F. Hübsch, N. Jung und S. Bräse, „Chemotion ELN: An Open Source electronic lab notebook for chemists in academia“, *Journal of Cheminformatics*, Jg. 9, Nr. 1, Dez. 2017. DOI: 10.1186/s13321-017-0240-0.

[95] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang und S. H. Bryant, „PubChem Substance and Compound databases“, *Nucleic Acids Research*, Jg. 44, Nr. D1, S. D1202–D1213, 4. Jan. 2016. DOI: 10.1093/nar/gkv951.

[96] P. Tremouilhac, C.-L. Lin, P.-C. Huang, Y.-C. Huang, A. Nguyen, N. Jung, F. Bach, R. Ulrich, B. Neumair, A. Streit und S. Bräse, „The Repository Chemotion: Infrastructure for Sustainable Research in Chemistry“, *Angewandte Chemie International Edition*, Jg. 59, Nr. 50, S. 22 771–22 778, 7. Dez. 2020. DOI: 10.1002/anie.202007702.

[97] European Organization For Nuclear Research und OpenAIRE, *Zenodo*, CERN, 2013. DOI: 10.25495/7GXK-RD71.

[98] Microsoft Corporation. „GitHub“. (2008), Adresse: https://github.com (eingesehen am 10. 08. 2023).

[99] G. King, „An Introduction to the Dataverse Network as an Infrastructure for Data Sharing“, *Sociological Methods & Research*, Jg. 36, Nr. 2, S. 173–199, Nov. 2007. DOI: 10.1177/0049124107306660.

[100] Harvard University. „Harvard Dataverse“. (2023), Adresse: https://dataverse.harvard.edu (eingesehen am 10. 08. 2023).

[101] H. Pampel, P. Vierkant, F. Scholze, R. Bertelmann, M. Kindling, J. Klump, H.-J. Goebelbecker, J. Gundlach, P. Schirmbacher und U. Dierolf, „Making Research Data Repositories Visible: The re3data.org Registry“, *PLoS ONE*, Jg. 8, Nr. 11, H. Suleman, Hrsg., e78080, 4. Nov. 2013. DOI: 10.1371/journal.pone.0078080.

[102] Deutsche Forschungsgemeinschaft. „DFG RIsources“. (2023), Adresse: https://risources.dfg.de (eingesehen am 10. 08. 2023).

[103] C. Draxl und M. Scheffler, „NOMAD: The FAIR concept for big data-driven materials science“, *MRS Bulletin*, Jg. 43, Nr. 9, S. 676–682, Sep. 2018. DOI: 10.1557/mrs.2018.208.

[104] The Galaxy Community, „The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update“, *Nucleic Acids Research*, Jg. 50, Nr. W1, W345–W351, 5. Juli 2022. DOI: 10.1093/nar/gkac247.

[105] S. P. Huber, S. Zoupanos, M. Uhrin, L. Talirz, L. Kahle, R. Häuselmann, D. Gresch, T. Müller, A. V. Yakutovich, C. W. Andersen, F. F. Ramirez, C. S. Adorf, F. Gargiulo, S. Kumbhar, E. Passaro, C. Johnston, A. Merkys, A. Cepellotti, N. Mounet, N. Marzari, B. Kozinsky und G. Pizzi, „AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance“, *Scientific Data*, Jg. 7, Nr. 1, S. 300, Dez. 2020. DOI: 10.1038/s41597-020-00638-4.

[106] Technische Informationsbibliothek (TIB). „TIB Terminology Service“. (2023), Adresse: https://terminology.tib.eu/ts (eingesehen am 10. 08. 2023).

[107] ORCID, Inc. „Open Researcher and Contributor ID“. (2012), Adresse: https://orcid.org (eingesehen am 10. 08. 2023).

[108] M. Hanke, F. Pestilli, A. S. Wagner, C. J. Markiewicz, J.-B. Poline und Y. O. Halchenko, „In defense of decentralized research data management“, *Neuroforum*, Jg. 27, Nr. 1, S. 17–25, 11. Jan. 2021. DOI: 10.1515/nf-2020-0037.

[109] M. R. Kennedy, „Nine questions to guide you in choosing a metadata schema“, *Journal of Digital Information*, Jg. 9, Nr. 1, 2008.

[110] E. Duval, W. Hodgins, S. Sutton und S. L. Weibel, „Metadata Principles and Practicalities“, *D-Lib Magazine*, Jg. 8, Nr. 4, Apr. 2002. DOI: 10.1045/april2002-weibel.

[111] R. Heery und M. Patel, „Application Profiles: Mixing and Matching Metadata Schemas“, *Ariadne*, Nr. 25, 2000, ISSN: 1361-3200.

[112] Publications Office of the European Union. „European Data Portal“. (2023), Adresse: https://data.europa.eu (eingesehen am 10. 08. 2023).

[113] F. Kirstein, B. Dittwald, S. Dutkowski, Y. Glikman, S. Schimmler und M. Hauswirth, „Linked Data in the European Data Portal: A Comprehensive Platform for Applying DCAT-AP“, in *Electronic Government*, I. Lindgren, M. Janssen, H. Lee, A. Polini, M. P. Rodríguez Bolívar, H. J. Scholl und E. Tambouris, Hrsg., Bd. 11685, Cham: Springer International Publishing, 2019, S. 192–204. DOI: 10.1007/978-3-030-27325-5_15.

[114] T. R. Bruce und D. I. Hillmann, „The continuum of metadata quality: defining, expressing, exploiting“, in *Metadata in Practice*, ALA editions, 2004.

[115] B. Stvilia, L. Gasser, M. B. Twidale und L. C. Smith, „A framework for information quality assessment“, *Journal of the American Society for Information Science and Technology*, Jg. 58, Nr. 12, S. 1720–1733, Okt. 2007. DOI: 10.1002/asi.20652.

[116] National Information Standards Organization, *A Framework of Guidance for Building Good Digital Collections*. Baltimore, MD: National Information Standards Organization (NISO), 2007, ISBN: 978-1-880124-74-1.

[117] J.-R. Park, „Metadata Quality in Digital Repositories: A Survey of the Current State of the Art“, *Cataloging & Classification Quarterly*, Jg. 47, Nr. 3-4, S. 213–228, 9. Apr. 2009. DOI: 10.1080/01639370902737240.

[118] B. Stvilia, L. Gasser und M. B. Twidale, „Metadata Quality Problems in Federated Collections:“ In *Challenges of Managing Information Quality in Service Organizations*, L. Al-Hakim, Hrsg., IGI Global, 2007, S. 154–186. DOI: 10.4018/978-1-59904-420-0.ch008.

[119] A. Quarati und J. E. Raffaghelli, „Do researchers use open research data? Exploring the relationships between usage trends and metadata quality across scientific disciplines from the Figshare case“, *Journal of*

Information Science, Jg. 48, Nr. 4, S. 423–448, 4. Okt. 2020. DOI: 10.1177/0165551520961048.

[120] A. Quarati, „Open Government Data: Usage trends and metadata quality“, *Journal of Information Science*, Jg. 49, Nr. 4, S. 887–910, 7. Okt. 2021. DOI: 10.1177/01655515211027775.

[121] D. R. Donaldson und J. W. Koepke, „A focus groups study on data sharing and research data management“, *Scientific Data*, Jg. 9, Nr. 1, S. 345, Dez. 2022. DOI: 10.1038/s41597-022-01428-w.

[122] J. Greenberg, M. C. Pattuelli, B. Parsia und W. D. Robertson, „Author-generated dublin core metadata for web resources: A baseline study in an organization“, *International Conference on Dublin Core and Metadata Applications*, S. 38–45, Okt. 2001.

[123] J. Greenberg, K. Spurgin und A. Crystal, „Final report for the AMeGA (automatic metadata generation applications) project“, 2005.

[124] M. Hatala und S. Forth, „A Comprehensive System for Computer-Aided Metadata Generation“, Jan. 2003.

[125] F. A. Musyaffa, K. Rapp und H. Gohlke, „LISTER: Semiautomatic Metadata Extraction from Annotated Experiment Documentation in eLabFTW“, *Journal of Chemical Information and Modeling*, acs.jcim.3c00744, 29. Sep. 2023. DOI: 10.1021/acs.jcim.3c00744.

[126] J. Barton, S. Currier und J. M. N. Hey, „Building quality assurance into metadata creation: An analysis based on the learning objects and e-Prints communities of practice“, *International Conference on Dublin Core and Metadata Applications*, S. 39–48, Sep. 2003.

[127] D. I. Hillmann, „Metadata Quality: From Evaluation to Augmentation“, *Cataloging & Classification Quarterly*, Jg. 46, Nr. 1, S. 65–80, März 2008. DOI: 10.1080/01639370802183008.

[128] M. Baca, „Practical Issues in Applying Metadata Schemas and Controlled Vocabularies to Cultural Heritage Information“, *Cataloging & Classification Quarterly*, Jg. 36, Nr. 3-4, S. 47–55, Juni 2003. DOI: 10.1300/J104v36n03_05.

[129] S. Ram und J. Liu, „Understanding the Semantics of Data Provenance to Support Active Conceptual Modeling“, in *Active Conceptual Modeling of Learning*, P. P. Chen und L. Y. Wong, Hrsg., Bd. 4512, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 17–29. DOI: 10.1007/978-3-540-77503-4_3.

[130] M. Schröder, S. Staehlke, P. Groth, J. B. Nebe, S. Spors und F. Krüger, „Structure-based knowledge acquisition from electronic lab notebooks for research data provenance documentation“, *Journal of Biomedical Semantics*, Jg. 13, Nr. 1, S. 4, Dez. 2022. DOI: 10.1186/s13326-021-00257-x.

[131] E. I. Saleh, „Image embedded metadata in cultural heritage digital collections on the web: An analytical study“, *Library Hi Tech*, Jg. 36, Nr. 2, S. 339–357, 9. Mai 2018. DOI: 10.1108/LHT-03-2017-0053.

[132] P. Bourhis, J. L. Reutter, F. Suárez und D. Vrgoč, „JSON: Data model, Query languages and Schema specification“, in *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, Chicago Illinois USA: ACM, 9. Mai 2017, S. 123–135. DOI: 10.1145/3034786.3056120.

[133] H. C. White, S. Carrier, A. Thompson, J. Greenberg und R. Scherle, „The dryad data repository: A singapore framework metadata architecture in a DSpace environment“, in *International Conference on Dublin Core and Metadata Applications*, Sep. 2008, S. 157–162.

[134] J. Greenberg, H. C. White, S. Carrier und R. Scherle, „A Metadata Best Practice for a Scientific Data Repository“, *Journal of Library Metadata*, Jg. 9, Nr. 3-4, S. 194–212, 30. Nov. 2009. DOI: 10.1080/19386380903405090.

[135] R. Duke, V. Bhat und C. Risko, „Data storage architectures to accelerate chemical discovery: Data accessibility for individual laboratories and the community“, *Chemical Science*, Jg. 13, S. 13 646–13 656, 2022. DOI: 10.1039/D2SC05142G.

[136] B. Heinrichs, N. Preuß, M. Politze, M. Müller und P. Pelz, „Automatic General Metadata Extraction and Mapping in an HDF5 Use-case“, in *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, SCITEPRESS - Science and Technology Publications, 2021, S. 172–179. DOI: 10.5220/0010654100003064.

[137] J. Hötzer, A. Reiter, H. Hierl, P. Steinmetz, M. Selzer und B. Nestler, „The parallel multi-physics phase-field framework Pace3D“, *Journal of Computational Science*, Jg. 26, S. 1–12, Mai 2018. DOI: 10.1016/j.jocs.2018.02.011.

[138] M. Kindling und D. Strecker, „Data Quality Assurance at Research Data Repositories“, *Data Science Journal*, Jg. 21, S. 18, 22. Nov. 2022. DOI: 10.5334/dsj-2022-018.

[139] M. Assante, L. Candela, D. Castelli und A. Tani, „Are Scientific Data Repositories Coping with Research Data Publishing?“, *Data Science Journal*, Jg. 15, S. 6, 26. Apr. 2016. DOI: 10.5334/dsj-2016-006.

[140] L. Cai und Y. Zhu, „The Challenges of Data Quality and Data Quality Assessment in the Big Data Era“, *Data Science Journal*, Jg. 14, S. 2, 22. Mai 2015. DOI: 10.5334/dsj-2015-002.

[141] M. Mesnier, G. Ganger und E. Riedel, „Object-based storage“, *IEEE Communications Magazine*, Jg. 41, Nr. 8, S. 84–90, Aug. 2003. DOI: 10.1109/MCOM.2003.1222722.

[142] Amazon Web Services, Inc. „Amazon S3“. (2006), Adresse: https://aws.amazon.com/s3 (eingesehen am 10. 08. 2023).

[143] Dropbox, Inc. „Dropbox“. (2007), Adresse: https://www.dropbox.com (eingesehen am 10. 08. 2023).

[144] M. R. Palankar, A. Iamnitchi, M. Ripeanu und S. Garfinkel, „Amazon S3 for science grids: A viable solution?“, in *Proceedings of the 2008 international workshop on Data-aware distributed computing*, Boston MA USA: ACM, 24. Juni 2008, S. 55–64. DOI: 10.1145/1383519.1383526.

[145] F. Gadban und J. Kunkel, „Analyzing the Performance of the S3 Object Storage API for HPC Workloads“, *Applied Sciences*, Jg. 11, Nr. 18, S. 8540, 14. Sep. 2021. DOI: 10.3390/app11188540.

[146] S. Simms, S. Jones, D. Mietchen und T. Miksa, „Machine-actionable data management plans (maDMPs)“, *Research Ideas and Outcomes*, Jg. 3, e13086, 5. Apr. 2017. DOI: 10.3897/rio.3.e13086.

[147] T. Miksa, S. Oblasser und A. Rauber, „Automating Research Data Management Using Machine-Actionable Data Management Plans“, *ACM Transactions on Management Information Systems*, Jg. 13, Nr. 2, S. 1–22, 30. Juni 2022. DOI: 10.1145/3490396.

[148] C. D. Hawker, „Laboratory Automation: Total and Subtotal“, *Clinics in Laboratory Medicine*, Jg. 27, Nr. 4, S. 749–770, Dez. 2007. DOI: 10.1016/j.cll.2007.07.010.

[149] I. M. Pendleton, G. Cattabriga, Z. Li, M. A. Najeeb, S. A. Friedler, A. J. Norquist, E. M. Chan und J. Schrier, „Experiment Specification, Capture and Laboratory Automation Technology (ESCALATE): A software pipeline for automated chemical experimentation and data management“, *MRS Communications*, Jg. 9, Nr. 3, S. 846–859, Sep. 2019. DOI: 10.1557/mrc.2019.72.

[150] N. Brandt, N. T. Garabedian, E. Schoof, P. J. Schreiber, P. Zschumme, C. Greiner und M. Selzer, „Managing FAIR Tribological Data Using Kadi4Mat“, *Data*, Jg. 7, Nr. 2, S. 15, 25. Jan. 2022. DOI: 10.3390/data7020015.

[151] P. De Geest, F. Coppens, S. Soiland-Reyes, I. Eguinoa und S. Leo, „Enhancing RDM in Galaxy by integrating RO-Crate“, *Research Ideas and Outcomes*, Jg. 8, e95164, 12. Okt. 2022. DOI: 10.3897/rio.8.e95164.

[152] P. Wittenburg, I. Anders, C. Blanchi, M. Buurman, C. Goble, J. Grieb, A. Hardisty, S. Islam, T. Jejkal, T. Kálmán, C. Kirkpatrick, L. Lannom, T. Lauer, G. Manepalli, K. Peters-von Gehlen, A. Pfeil, R. Quick, M. van de Sanden, U. Schwardmann, S. Soiland-Reyes, R. Stotzka,

Z. Trautt, D. Van Uytvanck, C. Weiland und P. Wieder, „FAIR Digital Object Demonstrators 2021“, Zenodo, 18. Jan. 2022. DOI: 10.5281/ZENODO.5872645.

[153] S. Cantor und T. Scavo, „Shibboleth Architecture“, *Protocols and Profiles*, Jg. 10, S. 16, 2005.

[154] „Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0“, OASIS, OASIS Standard, 15. März 2005.

[155] Verein zur Förderung eines Deutschen Forschungsnetzes. „DFN-AAI“. (2023), Adresse: https://www.aai.dfn.de (eingesehen am 10. 08. 2023).

[156] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros und C. Mortimore, „OpenID Connect Core 1.0“, The OpenID Foundation, OpenID Connect Specification, Nov. 2014.

[157] D. Hardt, „The OAuth 2.0 Authorization Framework“, RFC Editor, RFC 6749, Okt. 2012, S. 1–76.

[158] W. Pempe, „Grundlagen: AAI, Web-SSO, Metadaten und Föderationen“, präsentiert auf DFN-AAI Workshop, Aug. 2018.

[159] J. Sermersheim, „Lightweight Directory Access Protocol (LDAP): The Protocol“, RFC Editor, RFC 4511, Juni 2006, S. 1–68.

[160] OpenLDAP Foundation. „OpenLDAP“. (1998), Adresse: https://www.openldap.org (eingesehen am 10. 08. 2023).

[161] D. F. Ferraiolo und D. R. Kuhn, „Role-Based Access Controls“, in *Proceedings of the 15th Annual Conference on National Computer Security*, 1992.

[162] „eXtensible Access Control Markup Language (XACML) Version 1.0“, OASIS, OASIS Standard, 18. Feb. 2003.

[163] R. Shirey, „Internet Security Glossary, Version 2“, RFC Editor, RFC 4949, Aug. 2007, S. 1–356.

[164] J. Barkley, „Comparing simple role based access control models and access control lists“, in *Proceedings of the second ACM workshop on Role-based access control*, 1997, S. 127–132.

[165] A. Carusi und T. Reimer, „Virtual Research Environment Collaborative Landscape Study“, *JISC Report*, Jan. 2010.

[166] N. Brandt, L. Griem, C. Herrmann, E. Schoof, G. Tosato, Y. Zhao, P. Zschumme und M. Selzer, „Kadi4Mat: A Research Data Infrastructure for Materials Science“, *Data Science Journal*, Jg. 20, S. 8, 10. Feb. 2021. DOI: 10.5334/dsj-2021-008.

[167] Kadi4Mat Team and Contributors, *kadi: 0.42.0*, Version 0.42.0, Zenodo, 24. Okt. 2023. DOI: 10.5281/ZENODO.10037114.

[168] Pallets Projects. „Flask“. (2010), Adresse: https://palletsprojects.com/p/flask (eingesehen am 10. 08. 2023).

[169] P. J. Eby, „Python Web Server Gateway Interface v1.0.1“, PEP 3333, 2010.

[170] Django Software Foundation. „Django“. (2005), Adresse: https://www.djangoproject.com (eingesehen am 10. 08. 2023).

[171] Celery Team and Contributors. „Celery“. (2009), Adresse: https://docs.celeryq.dev (eingesehen am 10. 08. 2023).

[172] Redis Ltd. „Redis“. (2009), Adresse: https://redis.io (eingesehen am 10. 08. 2023).

[173] ISO Central Secretary, „Date and time — Representations for information interchange — Part 1: Basic rules“, International Organization for Standardization, Geneva, CH, Standard ISO 8601-1:2019, Feb. 2019.

[174] A. Wright, H. Andrews, B. Hutton und G. Dennis, „JSON Schema: A Media Type for Describing JSON Documents“, Internet Engineering Task Force, IETF Standard, 10. Juni 2022.

[175] PostgreSQL Global Development Group. „PostgreSQL“. (1996), Adresse: https://www.postgresql.org (eingesehen am 10. 08. 2023).

[176] E. F. Codd, „A relational model of data for large shared data banks“, *Communications of the ACM*, Jg. 13, Nr. 6, S. 377–387, Juni 1970. DOI: 10.1145/362384.362685.

[177] M. Bayer, „SQLAlchemy“, in *The architecture of open source applications volume II: Structure, scale, and a few more fearless hacks*, A. Brown und G. Wilson, Hrsg., Mountain View: aosabook.org, 2012.

[178] Elastic N.V. „Elasticsearch“. (2010), Adresse: https://www.elastic.co/elasticsearch (eingesehen am 10. 08. 2023).

[179] The Apache Software Foundation. „Apache Lucene“. (1999), Adresse: https://lucene.apache.org (eingesehen am 10. 08. 2023).

[180] A. Savaris, G. Colonetti, R. de Melo und A. von Wangenheim, „Relational Databases versus Search Engines: A Performance Comparison for Storing and Querying DICOM Metadata“, in *Anais do XVI Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS 2016)*, Brasilien: Sociedade Brasileira de Computação - SBC, 4. Juli 2016, S. 2557–2566. DOI: 10.5753/sbcas.2016.9902.

[181] MinIO, Inc. „MinIO“. (2014), Adresse: https://min.io (eingesehen am 10. 08. 2023).

[182] Vue Core Development Team. „Vue.js“. (2014), Adresse: https://vuejs.org (eingesehen am 10. 08. 2023).

[183] Pallets Projects. „Jinja2“. (2008), Adresse: https://palletsprojects.com/p/jinja (eingesehen am 10. 08. 2023).

[184] M. Otto, J. Thornton und Bootstrap Core Team and Contributors. „Bootstrap“. (2011), Adresse: https://getbootstrap.com (eingesehen am 10. 08. 2023).

[185] R. T. Fielding, „Architectural styles and the design of network-based software architectures“, Diss., University of California, Irvine, CA, USA, 2000.

[186] The GraphQL Foundation. „GraphQL“. (2015), Adresse: https://graphql.org (eingesehen am 10. 08. 2023).

[187] Google LLC. „gRPC“. (2016), Adresse: https://grpc.io (eingesehen am 10. 08. 2023).

[188] Kadi4Mat Team and Contributors, *kadi-apy: 0.34.0*, Version 0.34.0, Zenodo, 25. Okt. 2023. DOI: 10.5281/ZENODO.10040938.

[189] H. Krekel. „pluggy“. (2015), Adresse: https://pluggy.readthedocs.io (eingesehen am 10. 08. 2023).

[190] C. Herrmann, S. Schmid, D. Schneider, M. Selzer und B. Nestler, „Computational Determination of Macroscopic Mechanical and Thermal Material Properties for Different Morphological Variants of Cast Iron“, *Metals*, Jg. 11, Nr. 10, S. 1588, 5. Okt. 2021. DOI: 10.3390/met11101588.

[191] R. Kulagin, P. Reiser, K. Truskovskyi, A. Koeppe, Y. Beygelzimer, Y. Estrin, P. Friederich und P. Gumbsch, „Lattice Metamaterials with Mesoscale Motifs: Exploration of Property Charts by Bayesian Optimization“, *Advanced Engineering Materials*, Jg. 25, S. 2 300 048, 28. März 2023. DOI: 10.1002/adem.202300048.

[192] N. T. Garabedian, P. J. Schreiber, N. Brandt, P. Zschumme, I. L. Blatter, A. Dollmann, C. Haug, D. Kümmel, Y. Li, F. Meyer, C. E. Morstein, J. S. Rau, M. Weber, J. Schneider, P. Gumbsch, M. Selzer und C. Greiner, „Generating FAIR research data in experimental tribology“, *Scientific Data*, Jg. 9, Nr. 1, S. 315, Dez. 2022. DOI: 10.1038/s41597-022-01429-9.

[193] F. Rahmanian, J. Flowers, D. Guevarra, M. Richter, M. Fichtner, P. Donnely, J. M. Gregoire und H. S. Stein, „Enabling Modular Autonomous Feedback-Loops in Materials Science through Hierarchical Experimental Laboratory Automation and Orchestration“, *Advanced Materials Interfaces*, Jg. 9, Nr. 8, S. 2 101 987, März 2022. DOI: 10.1002/admi.202101987.

[194] L. C. Griem, R. Thelen und M. Selzer, „Automated Documentation of Research Processes Using RDM“, *Proceedings of the Conference on Research Data Infrastructure*, Jg. 1, 7. Sep. 2023. DOI: 10.52825/cordi.v1i.411.

[195] B. Celik, R. Sandt, L. C. P. dos Santos und R. Spatschek, „Prediction of Battery Cycle Life Using Early-Cycle Data, Machine Learning and Data Management“, *Batteries*, Jg. 8, Nr. 12, S. 266, 1. Dez. 2022. DOI: 10.3390/batteries8120266.

[196] Y. Zhao, N. Schiffmann, A. Koeppe, N. Brandt, E. C. Bucharsky, K. G. Schell, M. Selzer und B. Nestler, „Machine Learning Assisted Design of Experiments for Solid State Electrolyte Lithium Aluminum Titanium Phosphate“, *Frontiers in Materials*, Jg. 9, S. 821 817, 3. Feb. 2022. DOI: 10.3389/fmats.2022.821817.

[197] D. Rajagopal, A. Koeppe, M. Esmaeilpour, M. Selzer, W. Wenzel, H. Stein und B. Nestler, „Data-Driven Virtual Material Analysis and Synthesis for Solid Electrolyte Interphases“, *Advanced Energy Materials*, S. 2 301 985, 12. Sep. 2023. DOI: 10.1002/aenm.202301985.

[198] Y. Zhao, S.-K. Otto, T. Lombardo, A. Henss, A. Koeppe, M. Selzer, J. Janek und B. Nestler, „Identification of Lithium Compounds on Surfaces of Lithium Metal Anode with Machine-Learning-Assisted Analysis of ToF-SIMS Spectra“, *ACS Applied Materials & Interfaces*, acsami.3c09643, 18. Okt. 2023. DOI: 10.1021/acsami.3c09643.

[199] B. Schmieg, N. Brandt, V. J. Schnepp, L. Radosevic, S. Gretzinger, M. Selzer und J. Hubbuch, „Structured Data Storage for Data-Driven Process Optimisation in Bioprinting“, *Applied Sciences*, Jg. 12, Nr. 15, S. 7728, 1. Aug. 2022. DOI: 10.3390/app12157728.

[200] D. Grijalva Garces, S. Strauß, S. Gretzinger, B. Schmieg, T. Jüngst, J. Groll, L. Meinel, I. Schmidt, H. Hartmann, K. Schenke-Layland, N. Brandt, M. Selzer, S. Zimmermann, P. Koltay, A. Southan, G. E. M. Tovar, S. Schmidt, A. Weber, T. Ahlfeld, M. Gelinsky, T. Scheibel, R. Detsch, A. R. Boccaccini, T. Naolou, C. Lee-Thedieck,

C. Willems, T. Groth, S. Allgeier, B. Köhler, T. Friedrich, H. Briesen, J. Buchholz, D. Paulus, A. Von Gladiss und J. Hubbuch, „On the reproducibility of extrusion-based bioprinting: round robin study on standardization in the field“, *Biofabrication*, Jg. 16, Nr. 1, S. 015 002, 11. Okt. 2023. DOI: 10.1088/1758-5090/acfe3b.

[201] Y. Zhao, S.-K. Otto, N. Brandt, M. Selzer und B. Nestler, „Application of Random Forests in ToF-SIMS Data“, *Procedia Computer Science*, Jg. 176, S. 410–419, 2020. DOI: 10.1016/j.procs.2020.08.042.

[202] A. Koeppe, F. Bamer, M. Selzer, B. Nestler und B. Markert, „Explainable Artificial Intelligence for Mechanics: Physics-Explaining Neural Networks for Constitutive Models“, *Frontiers in Materials*, Jg. 8, S. 824 958, 2. Feb. 2022. DOI: 10.3389/fmats.2021.824958.

[203] A. Koeppe, F. Bamer, M. Selzer, B. Nestler und B. Markert, „Workflow concepts to model nonlinear mechanics with computational intelligence“, *PAMM*, Jg. 21, Nr. 1, Dez. 2021. DOI: 10.1002/pamm.202100238.

[204] Y. Zhao, P. Altschuh, J. Santoki, L. Griem, G. Tosato, M. Selzer, A. Koeppe und B. Nestler, „Characterization of porous membranes using artificial neural networks“, *Acta Materialia*, Jg. 253, S. 118 922, Apr. 2023. DOI: 10.1016/j.actamat.2023.118922.

[205] Y. Meng, J. Xu, Z. Jin, B. Prakash und Y. Hu, „A review of recent advances in tribology“, *Friction*, Jg. 8, Nr. 2, S. 221–300, Apr. 2020. DOI: 10.1007/s40544-020-0367-2.

[206] E. Santner, „Computer support in tribology - experiments and database“, *Tribotest*, Jg. 2, Nr. 3, S. 267–280, März 1996. DOI: 10.1002/tt.3020020305.

[207] J. Rumble und L. Sibley, *Towards a tribology information system* (NBS Special Publication 737). National Bureau of Standards, 1987.

[208] M. Woydt, „Modern methods to retrieve innovative material solutions for tribosystems“, *Lubrication Engineering*, Jg. 56, S. 26–30, Mai 2000.

[209] T. D. B. Jacobs und L. Pastewka, „Surface topography as a material parameter“, *MRS Bulletin*, 31. Jan. 2023. DOI: 10.1557/s43577-022-00465-5.

[210] A. Vellore, S. Romero Garcia, D. A. Johnson und A. Martini, „Ambient and Nitrogen Environment Friction Data for Various Materials & Surface Treatments for Space Applications“, *Tribology Letters*, Jg. 69, Nr. 1, S. 10, März 2021. DOI: 10.1007/s11249-020-01391-w.

[211] A. Medina-Smith, C. A. Becker, R. L. Plante, L. M. Bartolo, A. Dima, J. A. Warren und R. J. Hanisch, „A Controlled Vocabulary and Metadata Schema for Materials Science Data Discovery“, *Data Science Journal*, Jg. 20, S. 18, 29. Apr. 2021. DOI: 10.5334/dsj-2021-018.

[212] P. Kügler, M. Marian, B. Schleich, S. Tremmel und S. Wartzack, „tribAIn—Towards an Explicit Specification of Shared Tribological Understanding“, *Applied Sciences*, Jg. 10, Nr. 13, S. 4421, 27. Juni 2020. DOI: 10.3390/app10134421.

[213] M. Manske und L. D. Crocker. „MediaWiki“. (2002), Adresse: https://www.mediawiki.org (eingesehen am 10. 08. 2023).

[214] M. A. Musen, „The protégé project: A look back and a look forward“, *AI Matters*, Jg. 1, Nr. 4, S. 4–12, 16. Juni 2015. DOI: 10.1145/2757001.2757003.

[215] N. Garabedian und I. Bagov, „TriboDataFAIR Ontology“, Version v0.1.1, 10. März 2023. DOI: 10.5281/ZENODO.7716129.

[216] I. Niles und A. Pease, „Towards a standard upper ontology“, in *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, Ogunquit Maine USA: ACM, 17. Okt. 2001, S. 2–9. DOI: 10.1145/505168.505170.

[217] L. N. Soldatova und R. D. King, „An ontology of scientific experiments“, *Journal of The Royal Society Interface*, Jg. 3, Nr. 11, S. 795–803, 22. Dez. 2006. DOI: 10.1098/rsif.2006.0134.

[218] M. Weber und N. Garabedian, *SurfTheOWL*, Version v0.1.0, Zenodo, 23. Nov. 2021. DOI: 10.5281/ZENODO.5720218.

[219] J.-B. Lamy, „Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies“, *Artificial Intelligence in Medicine*, Jg. 80, S. 11–28, Juli 2017. DOI: 10.1016/j.artmed.2017.07.002.

[220] R. Bitter, T. Mohiuddin und M. Nawrocki, *LabVIEW: Advanced programming techniques*, 2nd ed. Boca Raton, FL: CRC Press/Taylor & Francis Group, 2007, 499 S., ISBN: 978-0-8493-3325-5.

[221] N. Brandt, *FAIR Tribological Data Helper Scripts*, Version 1.0.0, Zenodo, 10. Dez. 2021. DOI: 10.5281/ZENODO.5772522.

[222] J. Ellson, E. Gansner, L. Koutsofios, S. C. North und G. Woodhull, „Graphviz— Open Source Graph Drawing Tools“, in *Graph Drawing*, P. Mutzel, M. Jünger und S. Leipert, Hrsg., bearb. von G. Goos, J. Hartmanis und J. van Leeuwen, Bd. 2265, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, S. 483–484. DOI: 10.1007/3-540-45848-4_57.

[223] N. Garabedian, P. Schreiber, Y. Li, I. Blatter, A. Dollmann, C. Haug, D. Kümmel, F. Meyer, C. Morstein, J. Rau und C. Greiner, *FAIR Data Package of a Tribological Showcase Pin-on-Disk Experiment*, Version 1.0.1, Zenodo, 23. Nov. 2021. DOI: 10.5281/ZENODO.6797703.

[224] I. Bagov, C. Greiner und N. Garabedian, „Collaborative Metadata Definition using Controlled Vocabularies, and Ontologies“, *Research Ideas and Outcomes*, Jg. 8, e94931, 12. Okt. 2022. DOI: 10.3897/rio.8.e94931.

[225] F. Xu, J. Celli, I. Rizvi, S. Moon, T. Hasan und U. Demirci, „A three-dimensional in vitro ovarian cancer coculture model using a high-throughput cell patterning platform“, *Biotechnology Journal*, Jg. 6, Nr. 2, S. 204–212, Feb. 2011. DOI: 10.1002/biot.201000340.

[226] D. Kilian, P. Sembdner, H. Bretschneider, T. Ahlfeld, L. Mika, J. Lützner, S. Holtzhausen, A. Lode, R. Stelzer und M. Gelinsky, „3D printing of patient-specific implants for osteochondral defects: Workflow for an

MRI-guided zonal design“, *Bio-Design and Manufacturing*, Jg. 4, Nr. 4, S. 818–832, Dez. 2021. DOI: 10.1007/s42242-021-00153-4.

[227] V. Mironov, T. Boland, T. Trusk, G. Forgacs und R. R. Markwald, „Organ printing: Computer-aided jet-based 3D tissue engineering“, *Trends in Biotechnology*, Jg. 21, Nr. 4, S. 157–161, Apr. 2003. DOI: 10.1016/S0167-7799(03)00033-7.

[228] P. S. Gungor-Ozkerim, I. Inci, Y. S. Zhang, A. Khademhosseini und M. R. Dokmeci, „Bioinks for 3D bioprinting: An overview“, *Biomaterials Science*, Jg. 6, Nr. 5, S. 915–946, 2018. DOI: 10.1039/C7BM00765E.

[229] E. Mancha Sánchez, J. C. Gómez-Blanco, E. López Nieto, J. G. Casado, A. Macías-García, M. A. Díaz Díez, J. P. Carrasco-Amador, D. Torrejón Martín, F. M. Sánchez-Margallo und J. B. Pagador, „Hydrogels for Bioprinting: A Systematic Review of Hydrogels Synthesis, Bioprinting Parameters, and Bioprinted Structures Behavior“, *Frontiers in Bioengineering and Biotechnology*, Jg. 8, S. 776, 6. Aug. 2020. DOI: 10.3389/fbioe.2020.00776.

[230] R. Kawalkar, H. K. Dubey und S. P. Lokhande, „A review for advancements in standardization for additive manufacturing“, *Materials Today: Proceedings*, Jg. 50, S. 1983–1990, 2022. DOI: 10.1016/j.matpr.2021.09.333.

[231] P. Li, A. Faulkner und N. Medcalf, „3D bioprinting in a 2D regulatory landscape: Gaps, uncertainties, and problems“, *Law, Innovation and Technology*, Jg. 12, Nr. 1, S. 1–29, 2. Jan. 2020. DOI: 10.1080/17579961.2020.1727054.

[232] A. Pössl, D. Hartzke, T. M. Schmidts, F. E. Runkel und P. Schlupp, „A targeted rheological bioink development guideline and its systematic correlation with printing behavior“, *Biofabrication*, Jg. 13, Nr. 3, S. 035 021, 1. Juli 2021. DOI: 10.1088/1758-5090/abde1e.

[233] A. A. Armstrong, A. G. Alleyne und A. J. Wagoner Johnson, „1D and 2D error assessment and correction for extrusion-based bioprinting using

process sensing and control strategies“, *Biofabrication*, Jg. 12, Nr. 4, S. 045 023, 1. Okt. 2020. DOI: 10.1088/1758-5090/aba8ee.

[234] H. Cao, S. Mushnoori, B. Higgins, C. Kollipara, A. Fermier, D. Hausner, S. Jha, R. Singh, M. Ierapetritou und R. Ramachandran, „A Systematic Framework for Data Management and Integration in a Continuous Pharmaceutical Manufacturing Processing Line“, *Processes*, Jg. 6, Nr. 5, S. 53, 10. Mai 2018. DOI: 10.3390/pr6050053.

[235] M. Di Prima, J. Coburn, D. Hwang, J. Kelly, A. Khairuzzaman und L. Ricles, „Additively manufactured medical products – the FDA perspective“, *3D Printing in Medicine*, Jg. 2, Nr. 1, S. 1, Dez. 2016. DOI: 10.1186/s41205-016-0005-9.

[236] M. Rimann, E. Bono, H. Annaheim, M. Bleisch und U. Graf-Hausner, „Standardized 3D Bioprinting of Soft Tissue Models with Human Primary Cells“, *SLAS Technology*, Jg. 21, Nr. 4, S. 496–509, Aug. 2016. DOI: 10.1177/2211068214567146.

[237] G. Gillispie, P. Prim, J. Copus, J. Fisher, A. G. Mikos, J. J. Yoo, A. Atala und S. J. Lee, „Assessment methodologies for extrusion-based bioink printability“, *Biofabrication*, Jg. 12, Nr. 2, S. 022 003, 19. Feb. 2020. DOI: 10.1088/1758-5090/ab6f0d.

[238] J. An, C. K. Chua und V. Mironov, „Application of Machine Learning in 3D Bioprinting: Focus on Development of Big Data and Digital Twin“, *International Journal of Bioprinting*, Jg. 7, Nr. 1, S. 342, 29. Jan. 2021. DOI: 10.18063/ijb.v7i1.342.

[239] The MathWorks, Inc. „MATLAB“. (2023), Adresse: https://www.mathworks.com/products/matlab.html (eingesehen am 10. 08. 2023).

[240] L. Griem, P. Zschumme, M. Laqua, N. Brandt, E. Schoof, P. Altschuh und M. Selzer, „KadiStudio: FAIR Modelling of Scientific Research Processes“, *Data Science Journal*, Jg. 21, S. 16, 23. Sep. 2022. DOI: 10.5334/dsj-2022-016.

[241] P. Zschumme, J. Steinhülb und N. Brandt, *process-manager: 0.6.0*, Version 0.6.0, Zenodo, 28. Sep. 2023. DOI: 10.5281/ZENODO.8385496.

[242] P. Zschumme, E. Schoof, L. Griem, M. Laqua, A. Koeppe und N. Brandt, *process-engine: 0.12.1*, Version 0.12.1, Zenodo, 28. Sep. 2023. DOI: 10.5281/ZENODO.8385429.

[243] Kadi4Mat Team and Contributors, *xmlhelpy: 0.13.0*, Version 0.13.0, Zenodo, 24. Okt. 2023. DOI: 10.5281/ZENODO.10036992.

[244] Pallets Projects. „Click“. (2014), Adresse: https://palletsprojects.com/p/click (eingesehen am 10. 08. 2023).

[245] Kadi4Mat Team and Contributors, *workflow-nodes: 0.20.0*, Version 0.20.0, Zenodo, 24. Okt. 2023. DOI: 10.5281/ZENODO.10037047.

[246] M. Flachmann, J. Biesinger, M. Gorenflo, I. Bagov, C. Greiner und N. Garabedian, *Tribological Experiments - Sapphire on Copper - FAIR Dataset*, Version 0.2.0, Zenodo, 11. Juni 2023. DOI: 10.5281/ZENODO.8024881.

[247] K. Lang, C. Assmann, N. Neute, R. Gerlach und J. Rex, „FAIR Assessment Tools Overview“, 27. Feb. 2023. DOI: 10.5281/ZENODO.7701941.

[248] GitLab, Inc. „GitLab“. (2011), Adresse: https://gitlab.com (eingesehen am 10. 08. 2023).

[249] L. Torvalds und J. Hamano. „git“. (2005), Adresse: https://git-scm.com (eingesehen am 10. 08. 2023).

[250] J.-r. Park und A. Brenza, „Evaluation of Semi-Automatic Metadata Generation Tools: A Survey of the Current State of the Art“, *Information Technology and Libraries*, Jg. 34, Nr. 3, S. 22–42, 24. Sep. 2015. DOI: 10.6017/ital.v34i3.5889.

[251] M. Weise, F. Kovacevic, N. Popper und A. Rauber, „OSSDIP: Open Source Secure Data Infrastructure and Processes Supporting Data Visiting“, *Data Science Journal*, Jg. 21, S. 4, 9. Feb. 2022. DOI: 10.5334/dsj-2022-004.

[252] „The ELN Consortium“. (2023), Adresse: https://github.com/TheELNConsortium (eingesehen am 10. 08. 2023).

[253] K. Biernacka, P. Buchholz, S. A. Danker, D. Dolzycka, C. Engelhardt, K. Helbig, J. Jacob, J. Neumann, C. Odebrecht, B. Petersen, B. Slowig, U. Trautwein-Bruns, C. Wiljes und U. Wuttke, „Train-the-Trainer-Konzept zum Thema Forschungsdatenmanagement“, Version 4, 11. Dez. 2021. DOI: 10.5281/ZENODO.5773203.

[254] C. Wiljes und P. Cimiano, „Teaching Research Data Management for Students“, *Data Science Journal*, Jg. 18, S. 38, 13. Aug. 2019. DOI: 10.5334/dsj-2019-038.

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AAI	Authentifizierungs- und Autorisierungsinfrastruktur
ABAC	Attribute Based Access Control
ACL	Access Control List
AP	Application Profile
API	Application Programming Interface
CLI	Command-Line Interface
DCAT	Data Catalog Vocabulary
DCC	Digital Curation Centre
DCMI	Dublin Core Metadata Initiative
DFG	Deutsche Forschungsgemeinschaft
DFN	Deutsches Forschungsnetz
DMP	Datenmanagementplan
DOI	Digital Object Identifier
EAV	Entity-Attribute-Value
EDP	European Data Portal
ELN	Electronic Lab Notebook
EOSC	European Open Science Cloud
EVA	Eingabe, Verarbeitung, Ausgabe
FAIR	Findable, Accessible, Interoperable, Reusable
FDM	Forschungsdatenmanagement

FDO	FAIR Digital Object
GUI	Graphical User Interface
HATEOAS	Hypermedia as the Engine of Application State
HPC	High Performance Computing
HTTP	Hypertext Transfer Protocol
IAM	Institut für Angewandte Materialien
IMT	Institut für Mikrostrukturtechnik
IRI	Internationalized Resource Identifier
Kadi4Mat	Karlsruher Dateninfrastruktur für die Materialwissenschaften
KIT	Karlsruher Institut für Technologie
LDAP	Lightweight Directory Access Protocol
LDP	Linked Data Platform
LIMS	Labor-Informations- und Management-System
NFDI	Nationale Forschungsdateninfrastruktur
NoSQL	Not only SQL
OIDC	OpenID Connect
OPM	Open Provenance Model
ORCID	Open Researcher and Contributor ID
ORM	Object-Relational Mapping
OWL	Web Ontology Language
PAT	Personal Access Token
PID	Persistent Identifier
RBAC	Role Based Access Control
RDA	Research Data Alliance
RDBMS	Relational Database Management System
RDF	Resource Description Framework
RDFS	RDF Schema

RDMO	Research Data Management Organiser
REST	Representational State Transfer
SAML	Security Assertion Markup Language
SOP	Standard Operating Procedure
SPARQL	SPARQL Protocol And RDF Query Language
SPA	Single-Page Application
SQL	Structured Query Language
SSO	Single Sign-on
TIB	Technische Informationsbibliothek
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VFU	Virtuelle Forschungsumgebung
W3C	World Wide Web Consortium
WSGI	Web Server Gateway Interface
XACML	eXtensible Access Control Markup Language
XMP	Extensible Metadata Platform

Eigene Publikationen

Als Erstautor

N. Brandt, L. Griem, C. Herrmann, E. Schoof, G. Tosato, Y. Zhao, P. Zschumme und M. Selzer, „Kadi4Mat: A Research Data Infrastructure for Materials Science“, *Data Science Journal*, Jg. 20, S. 8, 10. Feb. 2021. DOI: 10.5334/dsj-2021-008.

N. Brandt, N. T. Garabedian, E. Schoof, P. J. Schreiber, P. Zschumme, C. Greiner und M. Selzer, „Managing FAIR Tribological Data Using Kadi4Mat“, *Data*, Jg. 7, Nr. 2, S. 15, 25. Jan. 2022. DOI: 10.3390/data7020015.

B. Schmieg, N. Brandt, V. J. Schnepp, L. Radosevic, S. Gretzinger, M. Selzer und J. Hubbuch, „Structured Data Storage for Data-Driven Process Optimisation in Bioprinting“, *Applied Sciences*, Jg. 12, Nr. 15, S. 7728, 1. Aug. 2022. DOI: 10.3390/app12157728.

Als Co-Autor

Y. Zhao, S.-K. Otto, N. Brandt, M. Selzer und B. Nestler, „Application of Random Forests in ToF-SIMS Data“, *Procedia Computer Science*, Jg. 176, S. 410–419, 2020. DOI: 10.1016/j.procs.2020.08.042.

Y. Zhao, N. Schiffmann, A. Koeppe, N. Brandt, E. C. Bucharsky, K. G. Schell, M. Selzer und B. Nestler, „Machine Learning Assisted Design of Experiments for Solid State Electrolyte Lithium Aluminum Titanium Phosphate“, *Frontiers in Materials*, Jg. 9, S. 821 817, 3. Feb. 2022. DOI: 10.3389/fmats.2022.821817.

L. Griem, P. Zschumme, M. Laqua, N. Brandt, E. Schoof, P. Altschuh und M. Selzer, „KadiStudio: FAIR Modelling of Scientific Research Processes“, *Data Science Journal*, Jg. 21, S. 16, 23. Sep. 2022. DOI: 10.5334/dsj-2022-016.

N. T. Garabedian, P. J. Schreiber, N. Brandt, P. Zschumme, I. L. Blatter, A. Dollmann, C. Haug, D. Kümmel, Y. Li, F. Meyer, C. E. Morstein, J. S. Rau, M. Weber, J. Schneider, P. Gumbsch, M. Selzer und C. Greiner, „Generating FAIR research data in experimental tribology“, *Scientific Data*, Jg. 9, Nr. 1, S. 315, Dez. 2022. DOI: 10.1038/s41597-022-01429-9.

D. Grijalva Garces, S. Strauß, S. Gretzinger, B. Schmieg, T. Jüngst, J. Groll, L. Meinel, I. Schmidt, H. Hartmann, K. Schenke-Layland, N. Brandt, M. Selzer, S. Zimmermann, P. Koltay, A. Southan, G. E. M. Tovar, S. Schmidt, A. Weber, T. Ahlfeld, M. Gelinsky, T. Scheibel, R. Detsch, A. R. Boccaccini, T. Naolou, C. Lee-Thedieck, C. Willems, T. Groth, S. Allgeier, B. Köhler, T. Friedrich, H. Briesen, J. Buchholz, D. Paulus, A. Von Gladiss und J. Hubbuch, „On the reproducibility of extrusion-based bioprinting: round robin study on standardization in the field“, *Biofabrication*, Jg. 16, Nr. 1, S. 015002, 11. Okt. 2023. DOI: 10.1088/1758-5090/acfe3b.

A Anhang

A.1 Metriken zur Evaluation der FAIRness von Forschungsdaten

In diesem Abschnitt werden die FAIRsFAIR Data Object Assessment Metrics [37] auf die Funktionalitäten von Kadi4Mat angewandt, um eine qualitative Evaluation des Systems entsprechend der in Abschnitt 7.2 erläuterten Zielsetzung vorzunehmen. Für jede Metrik wird deren Bezeichner und Name aufgeführt, gefolgt von einer textuellen Beschreibung der jeweiligen Ergebnisse. Die im aktuellen Stand der Metriken definierten Konformitätsstufen werden absichtlich nicht angegeben, da diese nicht für alle Metriken verfügbar sind und ihr Schwerpunkt hauptsächlich auf der Bewertung von Repositorien im Rahmen publizierter Forschungsdaten liegt.

Die Bezeichner der einzelnen Metriken sind wie folgt aufgebaut, wobei die erste Metrik **FsF-F1-01D** als Beispiel dient:

- **FsF**: FAIRsFAIR
- **F1**: FAIR-Prinzip, in diesem Beispiel der erste Teilabschnitt des Findable-Prinzips
- **01**: Lokale ID im Falle von mehreren, mit einem FAIR-Prinzip verknüpften Metriken
- **D**: Daten (D) und/oder Metadaten (M)

FsF-F1-01D: Daten wird ein global eindeutiger Identifikator zugewiesen

Allen innerhalb von Kadi4Mat verwalteten Ressourcen wird ein automatisch generierter Identifikator (ID) entsprechend des Basisschemas für Metadaten zugewiesen, der innerhalb einer konkreten Instanz von Kadi4Mat eindeutig und nicht änderbar ist. Dies gilt nicht nur für Records als Ganzes, sondern insbesondere auch für individuelle Dateien einzelner Records. Durch die passende Kombination eines Identifikators mit der entsprechenden Basis-URL einer Kadi4Mat-Instanz wird die globale Eindeutigkeit der entstehenden URLs sichergestellt. Unter Verwendung von HTTP-Clients, wie z. B. Webbrowsern, sind diese zudem auflösbar.

FsF-F1-02D: Daten wird ein persistenter Identifikator zugewiesen

Während URLs wie in FsF-F1-01D global eindeutig sind, ist deren Persistenz nicht automatisch sichergestellt, da sich sowohl die Basis-URL einer Instanz von Kadi4Mat als auch die zugrunde liegenden Ressourcen ändern können. Unter Verwendung der integrierten Publizierungsschnittstelle können jedoch in Kadi4Mat verwaltete Forschungsdaten mit einem entsprechenden, persistenten DOI versehen werden. Dieser wiederum ist typischerweise in Form einer URL auflösbar und leitet z. B. im Fall von Zenodo auf eine entsprechende Übersichtsseite weiter.

FsF-F2-01M: Metadaten enthalten beschreibende Kernelemente (Ersteller, Titel, Identifikator, Herausgeber, Veröffentlichungsdatum, Zusammenfassung und Schlüsselwörter), um die Auffindbarkeit der Daten zu unterstützen

Ein Großteil der gelisteten Metadatenelemente (Ersteller, Titel, Zusammenfassung, Schlüsselwörter) sind bereits in vergleichbarer Form im Basisschema von Kadi4Mat enthalten. Bibliografische Metadaten wie der Herausgeber und das Veröffentlichungsdatum sind dagegen hauptsächlich bei der Publizierung von

Forschungsdaten und entsprechender Metadaten relevant, lassen sich prinzipiell jedoch als Teil der generischen Record-Metadaten hinterlegen. Für den Identifikator gelten die bereits in FsF-F1-01D und FsF-F1-02D erläuterten Aspekte. Um die Auffindbarkeit von Daten auch bei der Publizierung in Repositorien zu unterstützen, wird im Beispiel von Zenodo ein Großteil dieser Metadaten auf das entsprechende Metadatenschema abgebildet. Ein Crosswalk generischer Metadaten könnte jedoch zukünftig ebenfalls in Betracht gezogen werden, wobei sich die Verwendung standardisierter Terme in Form von IRIs zur eindeutigen Kennung entsprechender Metadatenelemente eignen kann.

FsF-F3-01M: Metadaten enthalten den Identifikator der Daten, die sie beschreiben

Da Records in Kadi4Mat Daten und Metadaten in Form eines Containers logisch gruppieren, enthalten letztere immer den Identifikator sämtlicher zugehöriger Daten (siehe auch FsF-F1-01D). Ähnliches gilt bei Nutzung der Publizierungsschnittstelle von Kadi4Mat, wobei durch das verwendete RO-Crate-Format die Daten zusätzlich als Teil der enthaltenen JSON-LD-Metadatendatei innerhalb des entsprechenden Archivs identifiziert werden, auch wenn es sich in diesem Fall um keinen standardisierten Identifikator handelt.

FsF-F4-01M: Metadaten werden so angeboten, dass sie von Maschinen abgerufen werden können

Sämtliche innerhalb von Kadi4Mat verwalteten Metadaten werden in unterschiedlichen, maschinenlesbaren Formaten, z. B. in JSON oder einem RDF-basierten Format (siehe auch FsF-I1-01M), zum Export über entsprechende Endpunkte bereitgestellt (siehe auch FsF-A1-02M). Bei Nutzung der Publizierungsschnittstelle von Kadi4Mat werden im Beispiel von Zenodo lediglich die grundlegenden Metadaten (siehe auch FsF-F2-01M) automatisch in ähnlicher Form bereitgestellt und bei Registrierung eines DOIs (siehe auch FsF-F1-02D) zusätzlich in dem von

DataCite bereitgestellten Dienst hinterlegt. Dieser Aspekt kann für unterschiedliche, möglicherweise fachspezifische, Repositorien variieren.

FsF-A1-01M: Metadaten enthalten die Zugriffsstufe und die Zugriffsbedingungen der Daten

Diese Metrik spielt lediglich bei der Publizierung von Forschungsdaten und entsprechender Metadaten eine Rolle, da hier Aspekte wie Embargofristen oder Zugriffsbeschränkungen relevant sein können, welche über die Kollaboration mithilfe der Zugriffsrechteverwaltung von Kadi4Mat hinausgehen. Lizenzinformationen sind in diesem Kontext ebenfalls relevant, jedoch bereits separat in FsF-R1.1-01M ausgeführt.

FsF-A1-02M: Metadaten sind über ein standardisiertes Kommunikationsprotokoll zugänglich

Sämtliche innerhalb von Kadi4Mat verwalteten Ressourcen sowie entsprechende Metadaten sind über eine webbasierte HTTP-API abrufbar. Bei der Publizierung von Forschungsdaten hängt die Erfüllung dieser Metrik vom entsprechenden Repositorium ab, im Beispiel von Zenodo wird eine vergleichbare API zur Verfügung gestellt.

FsF-A1-03D: Daten sind über ein standardisiertes Kommunikationsprotokoll zugänglich

Siehe FsF-A1-02M.

FsF-A2-01M: Metadaten bleiben verfügbar, auch wenn die Daten nicht mehr verfügbar sind

Diese Metrik spielt hauptsächlich bei der Publizierung von Forschungsdaten und entsprechender Metadaten eine Rolle, da in diesem Fall deren Persistenz gewährleistet werden muss. Da bei der Nutzung von Repositorien wie Zenodo üblicherweise DOIs zum Einsatz kommen, kann durch die Verwendung der entsprechenden Publizierungsschnittstelle von Kadi4Mat der dauerhafte Zugriff auf Metadaten sichergestellt werden (siehe auch FsF-F1-02D).

FsF-I1-01M: Metadaten werden mithilfe einer formalen Wissensrepräsentationssprache dargestellt

Kadi4Mat unterstützt u. a. einen RDF-basierten Export von Metadaten im gängigen Serialisierungsformat Turtle, wodurch in Kombination mit FsF-I2-01M die semantische Interoperabilität sämtlicher Metadaten ermöglicht werden kann. Dieses Format wird ebenfalls bei Export oder Publizierung von RO-Crates in den entsprechenden Archiven genutzt, wobei die zusätzlich enthaltene JSON-LD-Metadatendatei ebenfalls die grundlegenden Metadaten des Basisschemas von Kadi4Mat enthält.

FsF-I2-01M: Metadaten nutzen semantische Ressourcen

Semantische Ressourcen können innerhalb generischer Record-Metadaten in Form von IRIs spezifiziert werden, um Terme standardisierter Vokabulare zu beschreiben. Diese werden ebenfalls innerhalb der in FsF-I1-01M beschriebenen Formate genutzt.

FsF-I3-01M: Metadaten enthalten Verknüpfungen zwischen den Daten und den zugehörigen Entitäten

Benutzerdefinierte Informationen zur Datenherkunft lassen sich mithilfe von Record-Links in Kadi4Mat spezifizieren (siehe auch FsF-R1.2-01M). Verknüpfungen auf externe Entitäten sind aktuell mithilfe dieser Funktionalität nicht möglich, werden jedoch von Repositorien wie Zenodo in Form von einfachen Identifikatoren unterstützt. Zur Abbildung solcher Arten von Metadaten könnte zukünftig ein ähnlich wie in FsF-F2-01M beschriebener Crosswalk in Kombination mit den generischen Record-Metadaten zum Einsatz kommen.

FsF-R1-01MD: Metadaten spezifizieren den Inhalt der Daten

Durch die Möglichkeit, anwendungsspezifische Informationen als Teil der generischen Record-Metadaten spezifizieren zu können, lassen sich die Inhalte von Forschungsdaten in beliebiger Ausführlichkeit beschreiben.

FsF-R1.1-01M: Metadaten enthalten Lizenzinformationen, unter welchen die Daten wiederverwendet werden können

Kadi4Mat ermöglicht die Spezifikation vordefinierter Lizenzen für individuelle Records, bei denen es sich größtenteils um standardisierte Lizenzen handelt. Diese sind ebenfalls als Teil der Metadaten in maschinenlesbarer Form abrufbar (siehe auch FsF-F4-01M) und können sowohl für den internen Datenaustausch als auch bei Nutzung der Publizierungsschnittstelle relevant sein. Im Beispiel von Zenodo können die hinterlegten Lizenzen in der Regel unverändert übernommen werden, da beide Systeme dieselben Arten von Lizenzen unterstützen.

FsF-R1.2-01M: Metadaten enthalten Herkunftsinformationen über die Erstellung oder Erzeugung von Daten

Mithilfe von Record-Links kann innerhalb von Kadi4Mat die vollständige Herkunft von Daten und allen darin beteiligen Prozessen und Akteuren spezifiziert werden. Diese sind ebenfalls als Teil der Record-Metadaten in unterschiedlichen Formaten und in maschinenlesbarer Form abrufbar (siehe auch FsF-F4-01M und FsF-I1-01M). Eine formale Beschreibung der Datenherkunft unter Verwendung von Ontologien wie PROV ist mithilfe eines entsprechenden Crosswalks prinzipiell ebenfalls möglich.

FsF-R1.3-01M: Metadaten liegen in einem von der angestrebten Forschungsgemeinschaft empfohlenen Standard vor

Die Erfüllung dieser Metrik hängt stark von der jeweiligen Forschungsdisziplin ab und lässt sich daher nicht in allgemeiner Form beantworten. Prinzipiell ermöglicht Kadi4Mat die Nutzung etablierter Metadatenschemata in Form generischer Record-Metadaten, semantischer Ressourcen (siehe auch FsF-I2-01M) und entsprechender Templates.

FsF-R1.3-02D: Daten liegen in einem von der angestrebten Forschungsgemeinschaft empfohlenen Dateiformat vor

Ähnlich wie FsF-R1.3-01M hängt die Erfüllung dieser Metrik von der jeweiligen Forschungsdisziplin ab, zusätzlich aber auch von verwendeter Software oder Gerätschaften, wodurch die Wahl der Dateiformate bereits im Vorfeld limitiert sein kann. Kadi4Mat schränkt daher die Wahl der Dateiformate nicht ein, bietet jedoch in Form von RO-Crates ein einheitliches Containerformat für Forschungsdaten.