FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Entwurf einer Architektur für maschinen-ausführbare Forschungsdatenmanagementpläne im institutionellen Kontext

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Simon Werner Oblasser, BSc
Matrikelnummer 01130928

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dr. Andreas Rauber
Mitwirkung: Dr. Tomasz Miksa

Wien, 4. Mai 2020

Simon Werner Oblasser	Andreas Rauber

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Designing an Architecture for Machine-actionable Research Data Management Planning in an Institutional Context

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Simon Werner Oblasser, BSc
Registration Number 01130928

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dr. Andreas Rauber
Assistance: Dr. Tomasz Miksa

Vienna, 4th May, 2020

Simon Werner Oblasser	Andreas Rauber

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Simon Werner Oblasser, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 4. Mai 2020

Simon Werner Oblasser

Danksagung

Mein tiefster Dank gilt meiner Verlobten Kathi für ihre Unterstützung und unendliche Geduld mit mir während dieser Reise und darüber hinaus.

Vielen Dank an meine Familie, ohne deren Liebe und Unterstützung diese Arbeit nicht möglich gewesen wäre.

Besonderer Dank gilt meinen Betreuern Andreas Rauber und Tomasz Miksa für ihren Rat und die Ermutigung nach Japan zu gehen.

Vielen Dank an Kitamoto-sensei, der mich als Praktikanten in sein Labor aufgenommen und mir die Freiheit gegeben hat, an dieser Arbeit und verwandten Themen zu arbeiten.

Vielen Dank an alle, die diese Arbeit ermöglicht haben, einschließlich der Mitglieder der Research Data Alliance DMP Common Standards-Arbeitsgruppe, des Zentrums für Forschungsdatenmanagement, der TU.it und der Campus-Softwareentwicklung an der TU Wien. Vielen Dank an meine ehemaligen Kolleginnen und Kollegen am Institut für Information Systems Engineering, insbesondere an Krešimir und Kabul für unzählige arbeitsbezogene und nicht arbeitsbezogene Diskussionen.

Acknowledgements

My deepest gratitude goes to my fiancee Kathi for her support and endless patience with me during this journey and beyond.

Many thanks to my family, without their love and support this thesis would not have been possible.

Special thanks to my advisors Andreas Rauber and Tomasz Miksa for their guidance and encouragement to go to Japan while working on this thesis.

Many thanks to Kitamoto-sensei, who accepted me as an intern in his laboratory and gave me the freedom to work on this thesis and related topics.

Thank you to everyone who made this work possible, including the members of the Research Data Alliance DMP Common Standards working group, the Center for Research Data Management, TU.it and Campus Software Development at the TU Wien. Thanks to my former colleagues at the Institute of Information Systems Engineering, especially to Krešimir and Kabul for countless work-related and non-work-related discussions.

Kurzfassung

Forschende sind zunehmend gefordert, Datenmangementpläne (DMPs) für Forschungsprojekte zu erstellen und zu pflegen, in denen beschrieben wird, wie Forschungsdaten verwaltet werden, um ihre Wiederverwendbarkeit sicherzustellen. Ein DMP, als statisches Dokument, kann schnell veraltet und unpraktisch zu warten sein. Daher wurde eine neue Generation von maschinen-ausführbaren DMPs (maDMPs) vorgeschlagen, welche es ermöglichen automatisiert Informationen zu integrieren und Aktualisierungen durchzuführen. MaDMPs eröffnen eine Vielzahl von Anwendungsfällen, welche die Interoperabilität von Forschungssystemen und die Automatisierung von Datenmanagement ermöglichen. In dieser Arbeit untersuchen wir maDMP-Anwendungsfälle von Stakeholdern innerhalb einer Forschungseinrichtung. Wir schlagen eine Systemarchitektur für die Unterstützung von maDMPs vor, welche in die Forschungsdatenmanagement-Infrastruktur einer Institution eingebettet ist. Um die Systemanforderungen zu definieren, verwenden wir grafische Modelle, um Feedback von Stakeholdern zu sammeln. Wir verwenden BPMN, um Geschäftsprozesse abzubilden, und die Modellierungssprache ArchiMate für die Beschreibung der Unternehmensarchitektur (Enterprise Architecture, EA). Wir machen eine Fallstudie am Beispiel der TU Wien und zeigen mit einer Proof-of-Concept-Implementierung, dass maDMP-Arbeitsabläufe halbautomatisiert werden können. Wir führen das Konzept eines Service Brokers in die Datenmanagementplanung ein, mithilfe dessen IKT-Dienste gefunden, konfiguriert und bereitgestellt werden können. Mit der Implementierung eines Service Brokers können Dienste der TU.it abstrahiert werden. Der aus dem Planungsprozess resultierende maDMP ist gemäß der Spezifikation des Research Data Alliance DMP Common Standards implementiert. Eine Evaluierung des maDMPs zeigt, dass er einen herkömmlichen DMP in Bezug auf Vollständigkeit und Ausdruckskraft nicht vollständig ersetzen kann, sondern diesen um maschinelle Ausführbarkeit erweitert.

Abstract

Researchers are increasingly required to create and maintain Data Management Plans (DMPs) for research projects that describe how research data is managed to ensure its reusability. A DMP being a static document can quickly become obsolete and impractical to maintain. A new generation of machine-actionable DMPs (maDMPs) was therefore proposed to enable automated integration of information and updates. MaDMPs open up a variety of use cases enabling interoperability of research systems and automation of data management tasks. In this work, we investigate maDMP use cases of stakeholders within a research institution. We propose an architecture of a maDMP support system embedded into the landscape of an institutional RDM infrastructure. To define requirements, we use graphical mockups to collect feedback from stakeholders. We use BPMN to model business processes and Enterprise Architecture (EA) modeling using the ArchiMate language to design the system. We conduct a case study using the example of the TU Wien and develop a proof-of-concept implementation which shows that maDMP workflows can be semi-automated. We introduce the concept of a service broker into data management planning which enables to discover, configure and provision ICT services. The implementation shows that services of the TU.it can be abstracted with the service broker. The maDMP resulting from the planning process is implemented in accordance with the specification of the Research Data Alliance DMP Common Standard. An evaluation of the maDMP shows that it cannot fully replace a conventional DMP in terms of completeness and expressiveness, but augments it with machine-actionability.

Contents

CHAPTER 1

Introduction

Today's scientific research is data-intensive and requires activities ranging from data collection, validation, curation and analysis to long-term preservation [HTT09]. Driven by the global developments towards Open Science [OEC15], research data including software have been established as first-class research products [HT20]. Access to research data is not only a prerequisite for the reproducibility of data-driven scientific experiments and the validation of scientific findings, but also enables data to be reused for new experiments and discoveries. The open access and sharing of research data from publicly funded research therefore increases the research potential of new digital technologies and networks and offers a higher return on public investement [OEC07].

The economic and societal benefits and increased effectiveness of research funding by sharing data prompted funding bodies and institutions to mandate researchers to write and maintain documents describing responsibilities for data sharing and management, known as Data Management Plans (DMPs) [SUDB18]. A DMP is a tool to demonstrate awareness of good data practice and is expected to outline how research data is created, managed, shared and preserved [Jon11]. A DMP typically describes, among other things, what types and volumes of research data will be created, how data will be organized and documented, how data quality will be assured, which kind of metadata will be created, how the data will be stored and backed-up during the project, how data will be shared, legal and ethical aspects, how data will be curated and preserved beyond the funding and what budget and resources are needed for implementing the plan [DCC13][Mic15]. However, depending on the funding organization or institution the requirements for a DMP differ in content and submission date and can therefore cover different phases of the data lifecycle [Mic15] [DCC] [WBZ17]. For example, the US National Science Foundation (NSF) requires a two pages DMP that must be submitted with the funding application [NSF18], while DMPs for research projects within the Horizon 2020 programme of the European Commission (EC) must be submitted within six month after the project starts and be updated for project reporting or when significant changes occur [Com16].

1.1 Motivation and Problem Statement

DMPs to date are free-form text documents that are created with text processors or online tools such as DMPonline[1] or DMPTool[2] [SSJR16] and can be exported as pdf, docx or similar file formats. The tools offer funder-specific DMP templates that mostly present open questions to the researcher and provide guidance texts depending on the themes[3] of the questions and the research organizations. The guidance is often perceived as too complicated, technical, vague or general, and researchers have problems answering the questions [GLJ+18]. The quality of the DMP therefore depends heavily on the data management knowledge of the authors. Unknowing researchers are unaware of the technical possibilities of suitable tools for managing research data, storage, repository systems or metadata standards and choose poor data management options. Ultimately DMPs may not describe the management of research data effectively and in sufficient quality or level of detail [SUDB18].

Researchers can perceive it as an administrative burden with limited use to write a static document often before the start of a project, rather than being an integral part of research practice [SJMM17]. Writing a DMP can be a tedious exercise since there do not exist any means of automation such as pre-filling a DMP with information from related information systems, an automated file analysis for the description of data, automated support for license selection, storage booking, data deposit, or cost estimation for data management resources [SJMM17] [MSMJ19]. Recipients of a DMP such as funders do not have automated support either and have to manually review them to check for the compliance with their data policy [SJMM17]. Other stakeholders involved in the process of Research Data Management (RDM), such as repository operators, are typically not involved in data management planning from the beginning in order to assist with data curation and enable a seamless transfer of data to their repositories at the end of a research project. ICT operators who provide computing resources and storage are not well integrated into data management planning and are not informed about upcoming service demands to plan their resources efficiently [SJMM17]. A major disadvantage arises from the current format of a DMP, since it is free-form text, the information it contains is not favorable for machine processing and therefore hinders to exploit the information for various automated use cases, such as monitoring, reporting or validation.

To sum up, a DMP in its current form does not reach its full potential as a satisfactory, integral part of research, which enables the flow of information and the automation of data management processes from which many stakeholders can benefit.

[1] https://dmponline.dcc.ac.uk
[2] https://dmptool.org
[3] https://github.com/DMPRoadmap/roadmap/wiki/Themes

1.2 Aim and Scope

Several international initiatives and organizations such as the Data Documentation Initiative (DDI) [AWG], the Research Data Alliance (RDA) [RDA15], the Australian National Data Service (ANDS)[4], the The Future of Research Communications and e-Scholarship (FORCE11)[5], or the EC [HJC+18] identified the limitations of current DMPs and suggest a development towards "active" DMPs [SJM+18]. A standardized machine-actionable (meta)data model [Gre13] [Fre16] [RDA17] for DMPs is a prerequisite for DMPs to become "active" and enable the exchange of DMP information between research systems and the automation of data management tasks throughout the research data lifecycle. The research community identified potential use cases [AWG] [SJ17] [SJMM17] [MSMJ19] and requirements [MNWR18] for machine-actionable Data Management Plans (maDMPs) and supporting systems. However, the use cases have not yet been realized and established in current data management practice.

In this work, we build on the community ideas and use cases around maDMPs and explore their feasibility in an institutional context such as a research institution or university with its systems, services and stakeholders. The aim of this work is to propose a system architecture for machine-actionable data management planning in an institutional context. This involves the

- identification of relevant stakeholders at a research institution and related organizations (e.g. funder) and their requirements for a machine-actionable data management planning support system.
- development and description of workflows / business processes for machine-actionable data management planning.
- description of the system architecture on an enterprise level that could serve as a basis for establishing a reference architecture for maDMPs at institutions.
- development of a proof-of-concept implementation that shows selected features of the described architecture.

The architecture design should describe how machine-actionable data management planning can be embedded into the landscape of an institutional research data management infrastructure. This involves the integration of services relevant for data management at the institution such as a Current Research Information System (CRIS), Information and Communications Technology (ICT) services and institutional archives, but also external information systems and services.

Many existing data management planning applications are awareness tools and focus on training researchers in good data management practice by guiding them through the

[4] https://www.ands.org.au/partners-and-communities/ands-communities/dmps-interest-group
[5] https://www.force11.org/group/fairdmp

process of data management planning with hints based on organizational requirements or guidelines set by experts like the Digital Curation Center (DCC). In contrast to that, the scope of this work is not on textual guidance for DMPs, but on establishing semi-automatic workflows for data management planning by means of system integration and automation to improve data management practice and facilitate the planning process. For that purpose services that provide structured machine-actionable information, which can be fed into a DMP and services that can consume this information and act upon it are required.

1.3 Research Questions

Given the motivation and aim of this work, we formulate the following research questions.

RQ1 What are the requirements for a machine-actionable data management planning support system in an institutional context?

(a) What are the workflows (tasks) of machine-actionable data management planning?

RQ2 What is a suitable architecture supporting machine-actionable data management planning at a research institution?

(a) How can the requirements be mapped into a modular architecture?

(b) What services are needed?

(c) Which services are institution-specific and which could be outsourced and shared with other institutions?

(d) How can we integrate the research data management services offered at the institution into data management planning?

RQ3 To what extent does the proposed system make the DMP process more efficient?

(a) Which tasks of data management planning can be supported with system integration and automation?

(b) Which degree of automation can we achieve for data management planning tasks?

RQ4 To what extent do the resulting DMPs meet the stakeholder requirements?

(a) To what extent can they follow the DMP templates of major funding organizations?

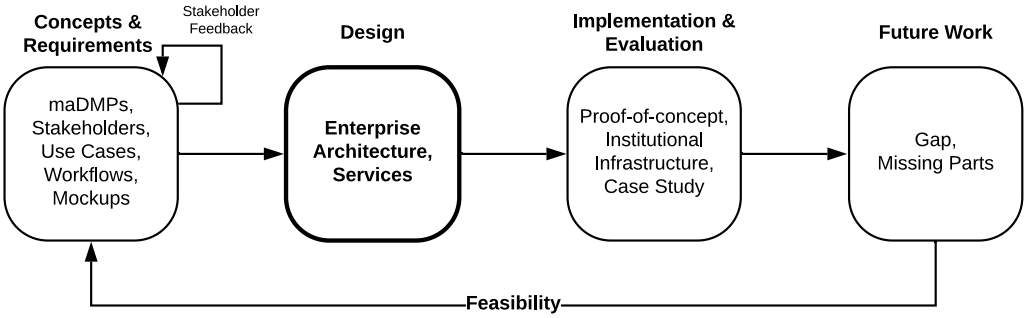


Figure 1.1: Overview of the methods used in this work to answer the research questions.

1.4 Method

We mostly use software engineering methods to pursue the aim of the work and to investigate the research questions. As a starting point, we study the scientific literature in order to identify the use cases for machine-actionable DMPs in an institutional context. Based on the relevant use cases, we model workflows of machine-actionable data management planning using a standard graphical notation for business process modeling. The workflows model the interactions between stakeholders and services involved in data management planning at an institution and can be understood by technical and non-technical people. In order to test the workflows, we design graphical mockups of the User Interfaces (UIs) for stakeholders involved in the workflows and present them to representatives of the stakeholder groups to further elicit requirements and collect feedback. Based on the feedback, the mockups and workflows are refined.

We then use the Enterprise Architecture (EA) modeling technique to design a machine-actionable DMP support system that can implement the business processes. The EA enables us to describe the system from different viewpoints and address stakeholder concerns. In the EA we describe the services needed for the implementation of the system.

We then create a case study at the example of the TU Wien and implement selected workflows and associated services of the architecture as a proof-of-concept, using the institutional infrastructure of systems and services of the TU Wien. We use the proof-of-concept implementation to evaluate the DMP efficiency gain by assessing the degree of automation achieved for data management planning tasks. We also assess the effectiveness with which the resulting maDMP can meet the stakeholder requirements. To evaluate how the maDMP meets funder requirements we use the DMP templates of two major funding bodies and assess to which extent they can be followed.

By using these methods, we can demonstrate the feasibility of machine-actionable DMP use cases and make statements about how well they could be realized with the proposed

system and which future developments are still required. Figure 1.1 gives an overview of the phases and research methods used in this thesis.

1.5 Thesis Overview

This section outlines the structure of this work and provides a preview of the content of the individual chapters.

- Chapter 2 provides the background needed to understand the broader context of the thesis and discusses related work. It describes the FAIR data principles and the lifecycle of research data, sketches a definition of machine-actionable DMPs and describes its current state-of-the-art, discusses the DMP tooling support provided today, and outlines the functionality of the Open Service Broker API (OSBAPI) - a standard mechanism for cloud service provisioning.
- Chapter 3 describes the requirements engineering carried out for this work. It provides an overview of the stakeholders and their use cases of an institutional machine-actionable DMP support system, describes our design of automated maDMP workflows and the requirements engineering we conducted with graphical UI mock-ups.
- Chapter 4 describes the proposed architecture for an institutional maDMP support system. It describes how the previously designed workflows of data management planning can be realized and which services are needed. It provides multiple concurrent views describing different aspects of the architecture design.
- Chapter 5 describes the proof-of-concept implementation of selected workflows and services of the proposed architecture in a real-world environment. It introduces the TU Wien case study and describes infrastructure and services relevant for data management at the TU Wien. It describes the implementation of DMap, a proof-of-concept tool that implements automated DMP workflows and connects to TU Wien services and external resources. It describes the implementation of the TU.it service broker, a tool that provides a standard interface to ICT services of the TU Wien.
- Chapter 6 describes the evaluation of the efficiency and effectiveness of the proposed system. It evaluates the degree of automation achieved for DMP tasks with the proof-of-concept implementation. It assesses how well the resulting maDMP meets stakeholder requirements, especially the requirements of funders, as these are the main recipients of DMPs.
- Chapter 7 describes the conclusions from this work and provides an outlook on future work. It revisits the research questions and describes how they were answered in this thesis. It shows limitations of this work and outlines future work.

CHAPTER 2

Background

In this chapter, we provide the background necessary to understand the broader context of the thesis and discuss related work. We describe the FAIR data principles, which are closely linked to good data management. We discuss the research data lifecycle models and the relation to DMPs. We sketch a definition of machine-actionable DMPs and describe the current state-of-the-art by outlining use cases for maDMPs, discussing principles for maDMPs and describing machine-actionable DMP data models. We discuss the DMP tooling support provided today and describe the OSBAPI - an open standard for cloud service provisioning.

2.1 FAIR Data

In order to effectively reuse research data, it must be Findable, Accessible, Interoperable, Reusable (FAIR) [WDA+16]. Findability, accessibility and interoperability of research data are prerequisites for its reusability [Mon18]. The FAIR guiding principles outline what is required for digital objects to be reusable not only for humans but also machines [WDA+16].

Findable To be findable a digital object must be assigned a unique, persistent identifier such as a Digital Object Identifier (DOI), have rich metadata attached and be indexed in a searchable resource.

Accessible To be accessible a digital object must be retrievable by its identifier in a standard way. Metadata should be accessible even when the actual data is not available. Access to data can be restricted though - FAIR data does not have to be open data [HBJ19].

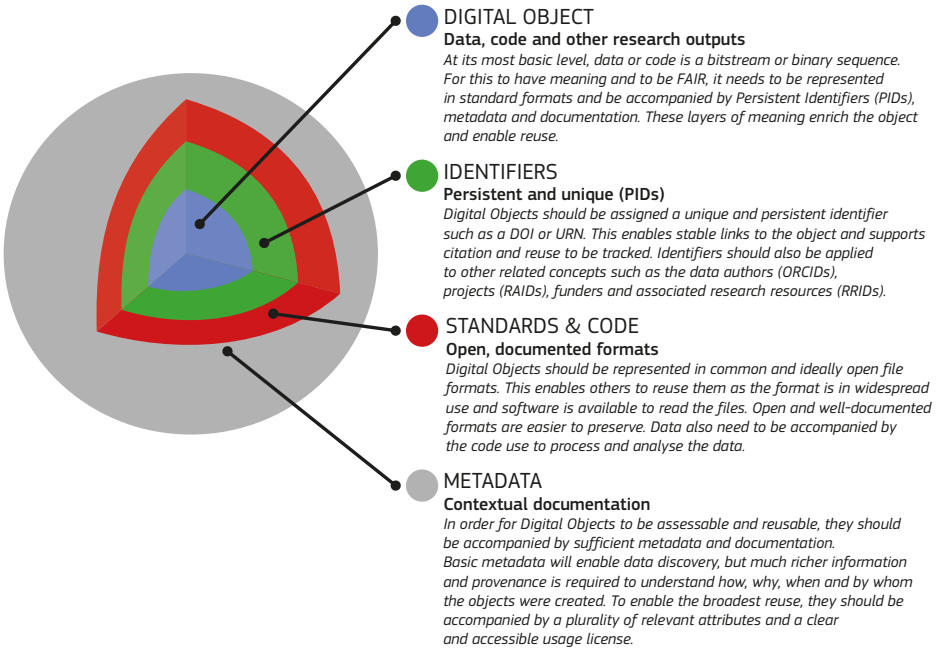


Figure 2.1: A model for FAIR Digital Objects [HJC+18].

Interoperable To be interoperable a digital object must use common formats, languages, vocabularies and include references to other data or metadata.

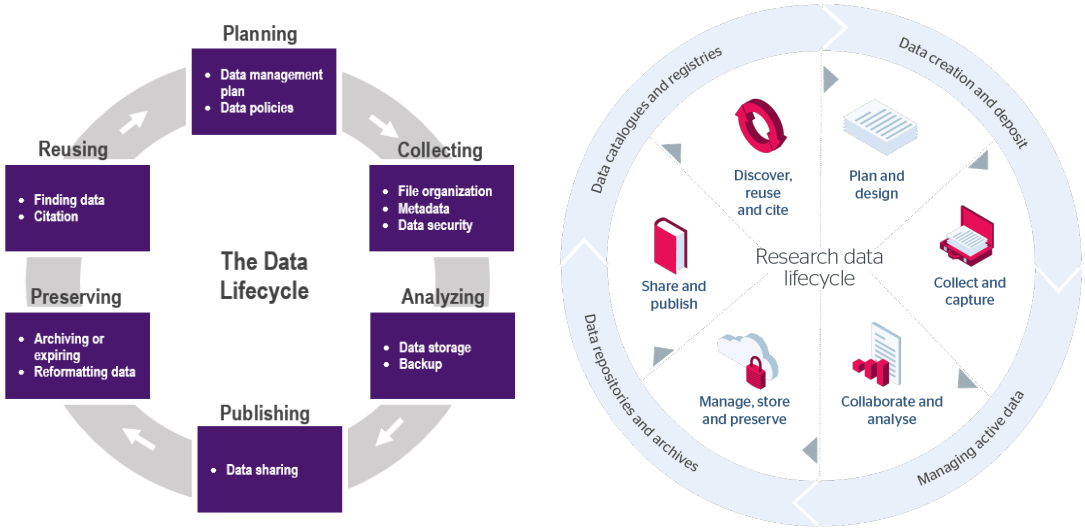
Reusable To be reusable a digital object must have accurate and extensive metadata, follow domain-relevant community standards, provide a provenance trace and have a clear usage license attached.

The FAIR data principles provide a measurable set of principles [WDA+16] that enable to assess the FAIRness of data [BDW19] [JG17]. The FAIR principles support good data management, therefore requirements for DMPs are increasingly aligned with the FAIR principles [Com16].

The layers of a FAIR digital object and associated elements such as identifiers, standards and metadata are depicted in Figure 2.1 [HJC+18].

2.2 Research Data Lifecycle

Research data often exceeds the lifespan of the research project from which it originated and must therefore be well managed to be reusable beyond funding [ECBW14]. Various data management activities are required during and after a research project to ensure data reuse. Data lifecycle models which extend the research lifecycle describe the phases that research data go through and the associated activities. Thus, activities from the



(a) Data lifecycle model[1].

(b) Jisc research data lifecycle (CC BY-ND) [Jisb].

Figure 2.2: Two similar lifecycle models, showing phases of the research data lifecycle and related data management activities.

domains of data curation, sharing, publishing and preserveration are integrated into the research process.

There exist different research data lifecycle models [Bal12] [MT18] [Car14]. Some lifecycle models focus on aspects of data curation and preservation [Hig08], others depict the research lifecycle within an organization [Lib], or visualize the broader range of research data management activities [Jisb] [Ser].

2.2.1 Phases

The research data lifecycle models depicted in Figure 2.2 describe typical phases of the research data lifecycle and related research data management activities. The following paragraphs give a brief summary of each phase [Ser] [Jisa].

Planning The central element in the planning phase is the DMP. It outlines how research data will be managed and shared in accordance with data policies of funders or institutions.

Collecting Research data is collected and captured from various sources. Active data needs to be stored and backed-up. In this phase, the organization of files, naming conventions and the capturing of metadata play a role.

[1] https://www.goshen.edu/academics/data-management/

Analyzing In this phase, active data is processed and analyzed. To enable the collaboration among team members active data needs to be shared. Data might be derived, checked, validated, cleaned, anonymized, described, documented and interpreted.

Publishing Data needs to be selected for sharing and publication, documented, enriched with discovery metadata and promoted. To enable data reuse, suitable licenses need to be assigned. Data can be shared in data repositories / archives to enable their discovery and access.

Preserving In this phase, data needs to be curated and preserved to enable authentic, long-term access to data. Data might be migrated to suitable formats.

Reusing Data can be reused for secondary analysis, follow-up research, reviews, the validation of findings or for educational purposes. Data catalogues and registries play a role in discovering and citing existing data.

2.2.2 DMP and Data Lifecycle

A DMP can cover the full data lifecycle and document data management activities for all phases of the lifcycle [Mic15]. However, Williams et al. [WBZ17] show that DMP requirements tend to focus on post-publication data sharing rather than preceding activities that affect data quality, ensure traceability, or support reproducibility.

A plan can change at any time or relevant information may not yet be available. Therefore, a best practice for funders is to require maintenance of DMPs after funding is granted [WBZ17] in order to make DMPs an evolving record of data management activities [HJC+18]. For example, the EC H2020 programme requires that a DMP is updated during a research project [Com16].

A DMP can be a powerful tool and help ensure data management quality when implemented as comprehensive documentation of the data lifecycle [WBZ17].

2.3 Machine-actionable DMPs

Machine-actionable DMPs are a vision for a new generation of data management plans. The central characteristic of a maDMP is that its information is modelled in a machine-actionable way. The term machine-actionable is associated with the FAIR principles to express that machines should be able to autonomously take action on digital objects [WDA+16]. Machine-actionability for DMPs shall be achieved by modelling the semantic information with the use of controlled vocabularies and standards [MNWR18]. Persistent Identifiers (PIDs) can be used to reference specific entities, such as people, institutions, funders, grants, datasets, or repositories [SJ17] [SJMM17].

Listing 2.1: Information of who is the creator of a DMP modelled in a machine-actionable way by using controlled vocabulary, standards and persistent identifiers [MNWR18].

```
"dc:creator": [{
        "foaf:name":"John Smith",
        "@id":"orcid.org/0000-1111-2222-3333",
        "foaf:mbox":"mailto:jsmith@tuwien.ac.at",
        "madmp:institution":"AT-Vienna-University-of-Technology"
}]
```

Listing 2.1 shows an example of how the information of who is the creator of a DMP can be modelled with the use of controlled vocabularies, standards and persistent identifiers [MNWR18]. In this example the information is expressed by using JSON-LD, existing vocabulary from the Dublin Core, and the ORCID persistent identifier. Wherever possible existing vocabulary should be reused since this is a best practice of the semantic web. However, the examples shows that custom vocabulary can be defined if necessary as it is the case with the institution.

When a DMP is machine-actionable, it can become a living (active) document [MSMJ19] since information can be integrated from various sources [MRGB17] and automatically be updated. Machine-actionable DMPs facilitate the exchange of information across research systems [SJMM17] and enable new use cases as described in the next section.

2.3.1 Community Use Cases

The community use cases for maDMPs described in this section form the basis of this work. In Chapter 3, we examine and deepen the use cases in the context of a research institution and its stakeholders and describe the associated requirements engineering carried out for this work.

In the following subsections, we summarize the results of various community efforts to collect use cases for maDMPs. In Section 2.3.1, we outline the community use cases resulting from a workshop at the International Digital Curation Conference (IDCC) in 2017. In Section 2.3.1, we describe the requirements engineering conducted by the Research Data Alliance DMP Common Standards working group. Section 2.3.1 discusses related results from a survey on the Horizon 2020 DMP template.

IDCC17 Workshop Results

At the 12th IDCC in February 2017, a workshop was organized by the DCC and the University of California Curation Center (UC3) to develop and prioritize use cases for maDMPs. The workshop methodology and results are described in [SJMM17]. Participants representing various stakeholders such as funders, developers, librarians, service providers and researchers developed use cases for defined topics and prioritized them in the following order.

1. Interoperability with other research systems
2. Leveraging persistent identifiers (PIDs)

3. Institutional use cases
4. Repository use cases
5. Data discovery and reuse
6. Evaluation and monitoring
7. Disciplinary tailoring and recommender systems
8. Publishing DMPs

The next few paragraphs summarize the identified use cases [SJMM17] for each of the topics. Note that some of the use cases overlap between topics.

Interoperability with other research systems The interoperability and information exchange between research systems, such as CRISs, funder systems, active storage, Electronic Lab Notebooks (ELNs), repositories, and publisher systems has been identified as the main topic for maDMP use cases. In addition to formulating the need for a common standard for the exchange of maDMP information, the idea of a common interface that would allow various tools and systems to interact with maDMPs has been described. For instance a workflow engine could provide provenance information or a file characterization tool could add identified file formats to the maDMP. Funders are the driving force behind DMPs. Therefore, an integration with funder systems for monitoring, reporting and compliance checking of maDMPs is highlighted.

Leveraging persistent identifiers PIDs have been identified to play a key role in maDMPs since they allow to make assertions about specific entities, such as researchers, funders, grants, organisations, repositories or datasets. Identified use cases for PIDs range from creating a dynamic inventory of research activities to triggering of notifications to automated reporting. In order to facilitate the researchers' administrative work, sections of a maDMP could be prefilled with PIDs from related information systems. Infrastructure providers mentioned in a maDMP could be notified with relevant information such as the expected volume, type of data and licensing.

Institutional use cases Challenges for institutions such as universities are related to connecting people, systems, resources and policies. MaDMPs could improve the situation by connecting people with data management services and tools, as well as helping institutions identify resource needs for better capacity planning. Use cases described include helping researchers select suitable IT services such as secure storage, High Performance Computing (HPC) or ELNs and integrating active data storage and data access management into maDMPs. Further use cases are connecting researchers with research support, e.g. ethics review or reporting, monitoring and compliance checking of institutional policies.

Repository use cases Identified repository use cases are a recommender system to find a suitable repository and a repository integration with maDMPs. Repository recommendation could be based on research discipline, country, type of data or other requirements, e.g. support of PIDs or trustworthiness. Sources for repository recommendation could be Data Cite's Registry of Research Data Repositories (re3data)[2] or the community-curated list from FAIRsharing[3]. Another approach than filtering these repository registries is to make recommendations from past repository selections mined from existing maDMPs. In this way popular repositories in a specific discipline or for a specific funder or institution could be recommended. An integration with repository systems could improve the data deposit process by informing repository managers about type of data, volume and standards early. Information from the maDMP could be used to populate metadata fields in the repository system when ingesting data and in return a DOI assigned to the dataset could be used to update the maDMP.

Data discovery and reuse Use cases of discoverability and reusability of maDMPs have been described. Published, versioned, machine-actionable DMPs enriched with PIDs present an opportunity to discover research and aggregate information spread beyond the maDMP. For instance maDMPs pointing to the same datasets would allow do discover related research. Information contained in related maDMPs can be a source of reuse when creating a new maDMP.

Evaluation and monitoring MaDMPs open up the possibility to automate the DMP review process with automated compliance checks. Stakeholders such as funders or institutions could get automated assistance in verifying if the maDMP was implemented. The quality and suitability of a maDMP in the respective discipline could be validated with automated support and relevant stakeholders be informed to provide assistance.

Disciplinary tailoring and recommender systems Data management practice can differ enormously between disciplines or subdisciplines. Providing suitable guidance is therefore a challenging task. Recommender systems, e.g. for standards or repositories as described above, could provide more tailored guidance.

Publishing DMPs As mentioned previously, public maDMPs enable the discovery of research and reuse of information. Additionally, public DMPs could serve as examples and training material for others. Public, versioned and updateable maDMPs could serve research projects throughout the lifecycle, not only at reporting phases.

[2] https://www.re3data.org/
[3] https://fairsharing.org/

RDA DMP Common Standard User Stories

The RDA DMP Common Standards working group[4] collected requirements for maDMPs [MNWR18]. The working group conducted an open stakeholder consultation to find out which stakeholders are involved at each stage of a research project's lifecycle, what information they need from a DMP, and who can provide that information at what time. For this purpose they collected user stories on GitHub[5] and during a workshop [MA17] in the form:

As a <stakeholder>, I want <goal> so that <reason>.

In this way, more than 100 user stories were collected and categorized according to stakeholders, phases of the lifecycle and DMP themes they are concerned with. The main goal for the working group was to derive requirements for a maDMP data model in order to find out which information should be contained in a common data model. However, many of the user stories provide an insight into the needs of stakeholder for a maDMP support system, since they are not only concerned with a maDMP data model, but also with the processes surrounding maDMPs.

Horizon 2020 DMP Survey

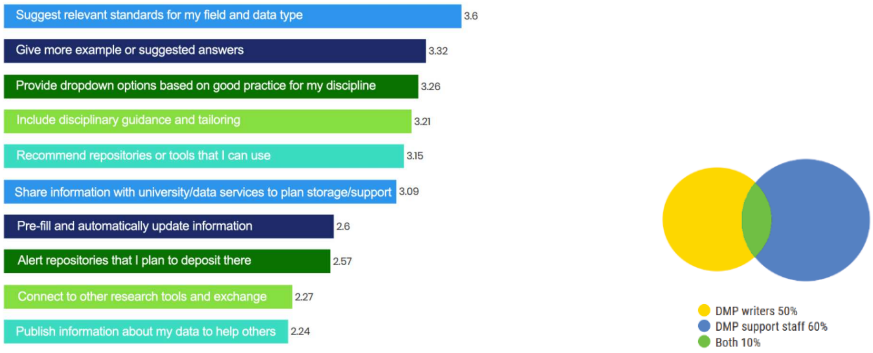
Within the EC OpenAIRE project a survey on the Horizon 2020 DMP template was conducted [GLJ+18]. The survey aimed to collect feedback from researchers and research support staff and gives valuable insights on their experience with the template. For instance 44% of survey participants agree that there are too many free text answers and more drop-down options are desirable.

In another question participants were ask to rank their top five out of ten priorities for a DMP template or tool, machine-actionable use cases were also among the options. Figure 2.3a shows the survey result for this question and indicates that users mainly want more suggestions, examples and guidance with respect to their field. Machine-actionable use cases like repository recommendation or sharing information with university or data services to plan storage/support were also ranked high, while other machine-actionable use cases were rated as less important. This result is not surprising since the survey participants are represented by researchers and research support staff only, but not by other stakeholders like institutions, service providers or funders who play a role in machine-actionable use cases, see Figure 2.3b.

2.3.2 Ten Principles

In order to make machine-actionable DMPs tangible a set of principles were proposed [MSMJ19]. The principles describe what is required to bring the maDMP vision to life and to realize use cases from stakeholders involved in research data management. Amongst other things, the principles suggest embedding maDMPs in the workflows of stakeholders

[4] https://www.rd-alliance.org/groups/dmp-common-standards-wg
[5] https://github.com/RDA-DMP-Common/user-stories



(a) Priorities for a DMP template or tool. (b) Roles of survey respondents.

Figure 2.3: Survey on H2020 DMP template [GLJ+18]. Survey respondents, being DMP writers and/or support staff wish for field or data type specific suggestions, examples, drop-down options, disciplinary guidance, repository and tools recommendations and to share information with university/data services to plan storage in a DMP template or tool.

to enable automation and to use a common data model for the exchange of DMP information. Further automation can be enabled by the machine-actionable description of resources such as data policies or components of the research data ecosystem such as repository systems. Figure 2.4 gives an overview of the principles for machine-actionable DMPs.

2.3.3 Data Model

A machine-actionable data model for DMPs makes it possible to express DMP information for humans and machines in a meaningful way. As a DMP information carrier, it forms the basis for interoperability between systems. Several initiatives described the need for a standard DMP data model or got engaged with active development.

Data Documentation Initiative The DDI, which hosts metadata standards to document the phases of the research data lifecycle, found that information related to data management planning could not be captured and suggested that this information be recorded in a standard manner [Gre13]. They developed a superset model for a DMP and assessed its mapping to their standard.

DataID DataID[6] which is an ontology that provides vocabulary to describe datasets in various forms, was extended to describe metadata related to DMPs [Fre16]. Therefore, they integrated an ontology based on the re3data schema [RVU+15] into their core

[6] http://dataid.dbpedia.org/ns/core.html

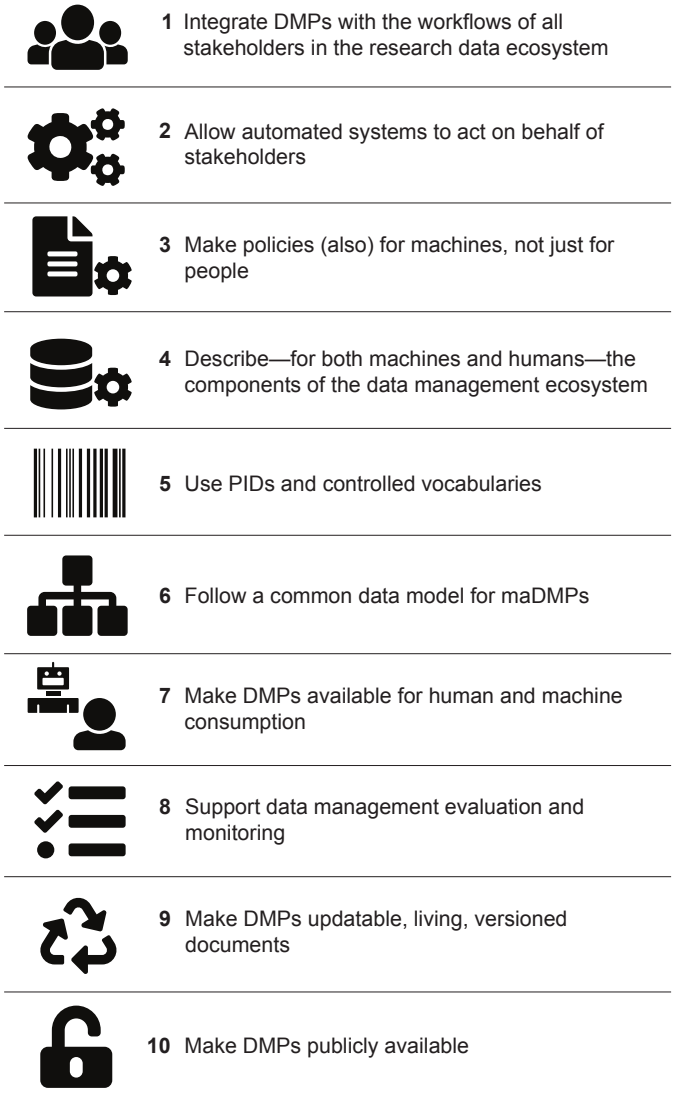


Figure 2.4: Ten principles for machine-actionable DMPs [MSMJ19].

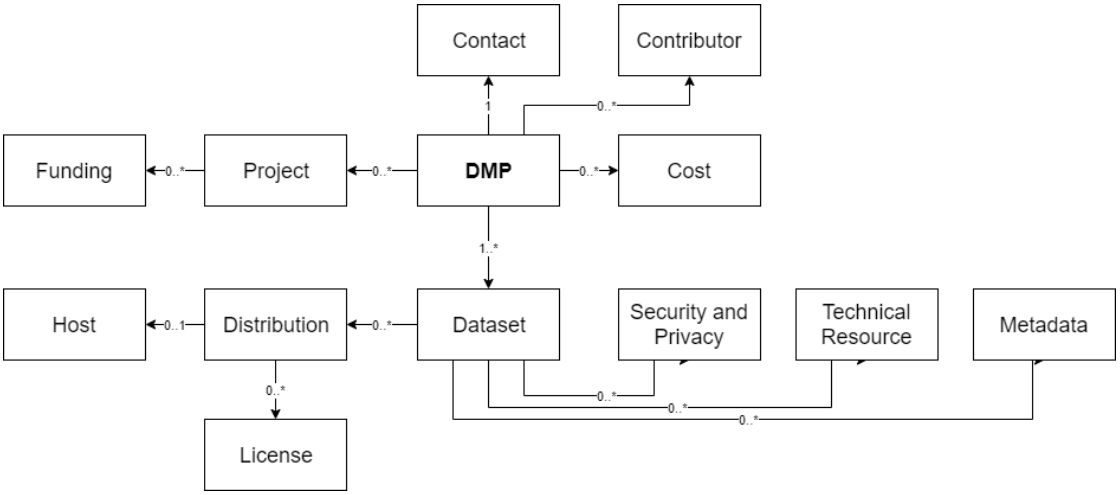


Figure 2.5: Overview of the RDA DMP Common Standard data model [MWN19].

ontology to describe repositories and institutions and made use of existing vocabularies such as DCAT, PROV-O and ORG [FBR+16].

Research Data Alliance The RDA DMP Common Standards working group [RDA17] [MNWR18] developed a common data model / metadata application profile for DMPs in JavaScript Object Notation (JSON) format [MWN19] and maintains a version represented in the Web Ontology Language (OWL). The RDA data model was developed as a collaborative effort by consulting stakeholders and collecting user stories as described in Section 2.3.1. It is capable of describing various entities involved in data management planning such as the DMP itself, projects, funding, contributors, costs, datasets and their relations. An overview of the RDA DMP Common Standard data model is given in Figure 2.5.

In this thesis, we rely on the work of the RDA DMP Common Standards working group and implement their data model in the proof-of-concept implementation, as described in Section 5.2.7.

2.4 DMP Tooling Support

DMPs are written in free-form text using checklists [DCC13], templates or tools that guide the planning process. The most popular online tools for creating a DMP are DMPonline[7] and DMPTool[8]. Their developers, the California Digital Library UC3 and the DCC, joined together and use the same code base under the DMPRoadmap[9] project [SJM+18]. These tools offer funder-specific DMP templates that mostly present open

[7] https://dmponline.dcc.ac.uk
[8] https://dmptool.org
[9] https://github.com/DMPRoadmap/roadmap

Table 2.1: Overview of current platforms and tools for data management planning. Adapted from [SJM+18] and [JPH+20].

DMP Platform/Tool	Organization(s)
ARGOS	OpenAIRE, EUDAT CDI
Data Stewardship Wizard (DSW)	Dutch Techcentre for Life Sciences (DTL, ELIXIR NL), Czech Technical University in Prague (CTU, ELIXIR CZ)
DataWiz	Leibniz Institute for Psychology Information and Documentation (ZPID)
DMPRoadmap (DMPonline, DMPTool, others)	California Digital Library (CDL), Digital Curation Centre (DCC), Portage Network, INIST CNRS
EasyDMP	EUDAT, UNINETT Sigma2
ezDMP	Interdisciplinary Earth Data Alliance (IEDA)
GFBio DMP Tool	The German Federation for Biological Data (GFBio)
ReDBox RDMP	Queensland Cyber Infrastructure Foundation
Research Data Management Organiser (RDMO)	Leibniz Institute for Astrophysics Potsdam (AIP), Potsdam University of Applied Sciences (FHP), Karlsruhe Institute of Technology Library (KIT)
University of Queensland Research Data Manager (UQRDM)	University of Queensland
Wizard for DMP Creation	CLARIN-D: Research Infrastructure for Humanities and Social Sciences

questions to the researcher and provide guidance texts depending on the themes[10] of the questions and the research organizations. The tools allow to collaboratively create DMPs, share and export them to various formats such as pdf, html or docx. DMPRoadmap is an open source project and can be adopted and branded by other institutions, for example DMPTuuli[11] from Finland or DMP OPIDoR[12] from France. In 2017 the California Digital Library received funding[13] from the NSF to develop DMPRoadmap with national and international collaborators towards machine-actionability.

There are many other tools for creating DMPs on the market, an overview of current platforms and tools for data management planning is given in Table 2.1 [SJM+18] [JPH+20]. The Data Stewardship Wizard[14] (DSW) is based on expert knowledge models from which questionnaires with different question types can be created [HKSP19].

[10] https://github.com/DMPRoadmap/roadmap/wiki/Themes
[11] https://www.dmptuuli.fi/
[12] https://dmp.opidor.fr/
[13] https://www.nsf.gov/awardsearch/showAward?AWD_ID=1745675
[14] https://ds-wizard.org/

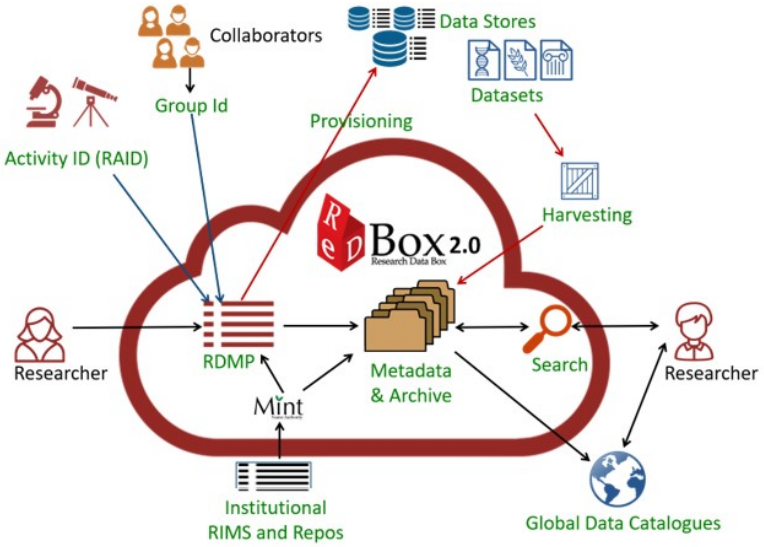
RDMOrganiser[15] (RDMO) offers similar functionalities like DMPonline and DMPTool and is to be deployed within RDM infrastructures of institutions in Germany, rather than hosting central instances on a national or international level. The University of Queensland Research Data Manager[16] (UQRDM) is a researcher-centric tool that follows an integrated approach and allows to actively manage data storage and access control. Data storage is automatically provisioned from the cloud computing platform QRIScloud[17] after creating a record in UQRDM. In this context, the term Data Management Record (DMR)[18] instead of DMP is used to emphasize that a DMR remains linked to the research data throughout the lifecycle of the project and relevant metadata is captured by the system, reducing the administrative burden on researchers. The Research Data Box[19] (ReDBox), also from Australia, is a research data management platform that connects research activities with research data, eResearch infrastructure and services for processing, storing and managing research data [Fou]. ReDBox includes a DMP tool which will be extended with a provisioner[20] service that allows researchers to create workspaces for self-provisioned research tools such as data storage, GitLab code repositories or LabArchive ELNs. An overview of the ReDBox 2.0 data management system is shown in Figure 2.6.

The design of the maDMP support system proposed in this work is influenced by ideas and developments of existing DMP tools such as DMPRoadmap and ReDBox RDMP.

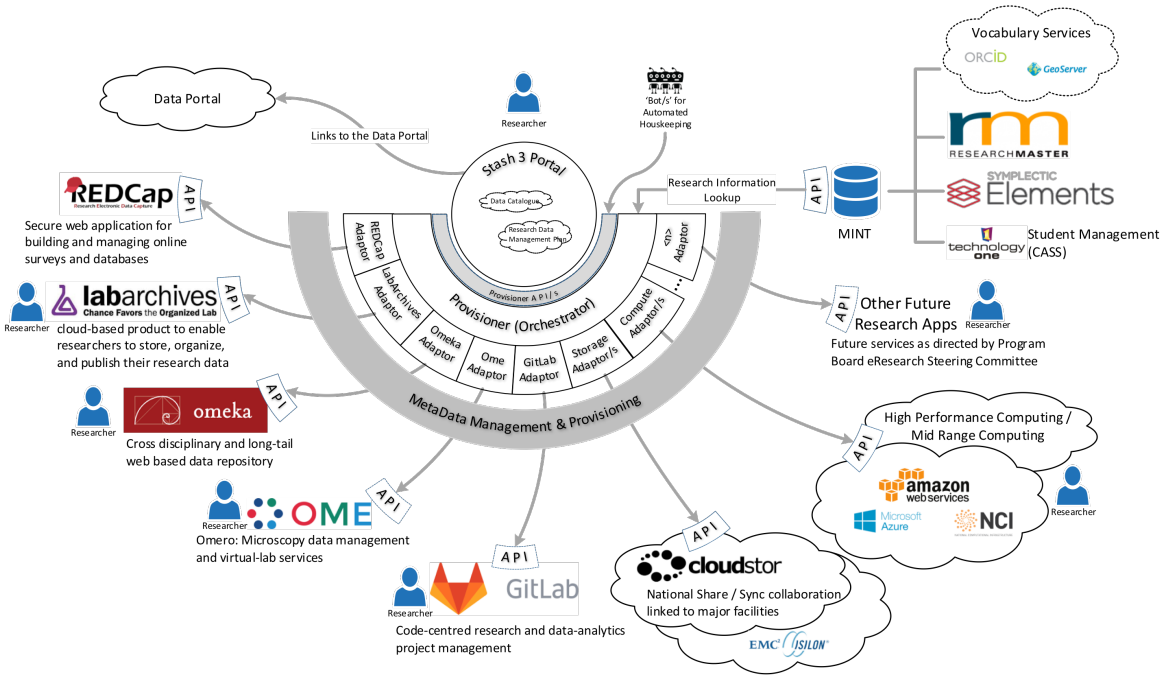
DMPRoadmap integrates the RDA Metadata Standards Directory[21] which inspired the automated DMP workflow for metadata standards selection, described in Section 3.2.6.

ReDBox 2.0 introduces “Provisioner”, an extension of the RDMP tool that integrates the provision of ICT services and research tools into data management planning and links the associated research workspaces with DMPs. The custom Provisioner Application Programming Interface (API) implements adaptors for the native APIs of the services, as shown in Figure 2.6b. In this work, however, we propose the use of service brokers following the OSBAPI, an open, industry-driven standard interface for cloud service provisioning, as described in Section 2.5. The OSBAPI is supported by a variety of service providers and enables the configuration, costing and provision of services in a standardized manner.

[15] https://rdmorganiser.github.io/en/
[16] https://research.uq.edu.au/project/research-data-manager-uqrdm
[17] https://www.qriscloud.org.au/
[18] https://rcc.uq.edu.au/article/2017/04/pilot-uq%E2%80%99s-innovative-research-data-management-system-underway
[19] https://www.redboxresearchdata.com.au/
[20] https://eresearch.uts.edu.au/2018/04/05/provisioner_1.htm
[21] https://rd-alliance.github.io/metadata-directory/

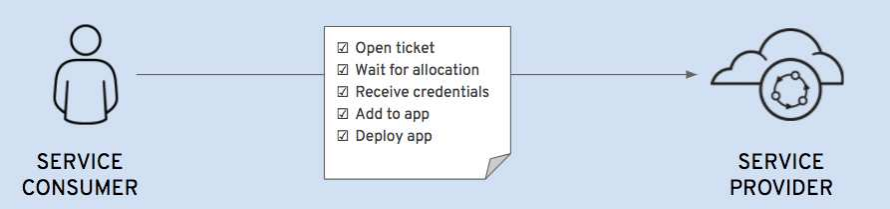


(a) ReDBox 2.0 data management system that supports the Research Data Lifecycle Framework.

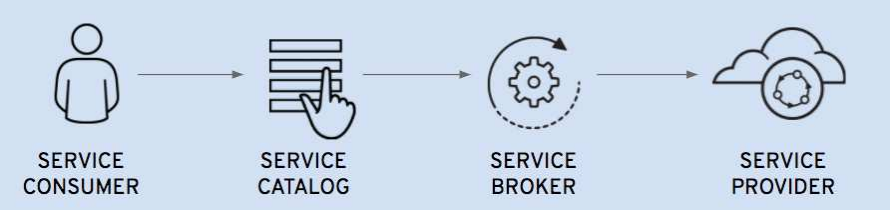


(b) Provisioner extends the RDMP tool in ReDBox 2.0 and enables researchers to request ICT services and research apps for data management and link the workspaces to DMPs.

Figure 2.6: ReDBox 2.0 data management system [Fou].



(a) Time consuming and error-prone process of manually requesting a service from a provider.



(b) Automated self-service for obtaining a service from a provider enabled by the OSBAPI.

Figure 2.7: Manual vs. automated approach of acquiring a service from a provider[25].

2.5 Open Service Broker API

The OSBAPI[22] is an industry-driven standard for cloud service provisioning across multiple service providers and developed by cloud companies such as Google, IBM, Pivotal, Red Hat, SAP. It enables to deliver services from different service providers to applications running on cloud platforms such as OpenShift, Kubernetes or Cloud Foundry.

Many cloud service providers such as Amazon Web Service[23] or Microsoft Azure[24] offer service brokers implementing the OSBAPI. A service broker implementing the OSBAPI can return a service catalog with different service plans that are associated with different Quality of Service (QoS) and cost. The service broker provides a standard interface for managing the lifecycle of a service instance from its creation at the service provider to its deletion.

Compared to a manual, often time consuming and error-prone process of acquiring a service from a provider, the OSBAPI enables to obtain a service in an automated self-service manner and drastically reduces the waiting time until the service can be used (Figure 2.7).

In this work, we use the concept of service brokers, implementing the OSBAPI for the integration of ICT services into data management planning. In a case study, we

[22] https://www.openservicebrokerapi.org/
[23] https://aws.amazon.com/partners/servicebroker/
[24] https://osba.sh/
[25] https://www.openshift.com/blog/whats-new-in-openshift-3-7-service-catalog-and-brokers

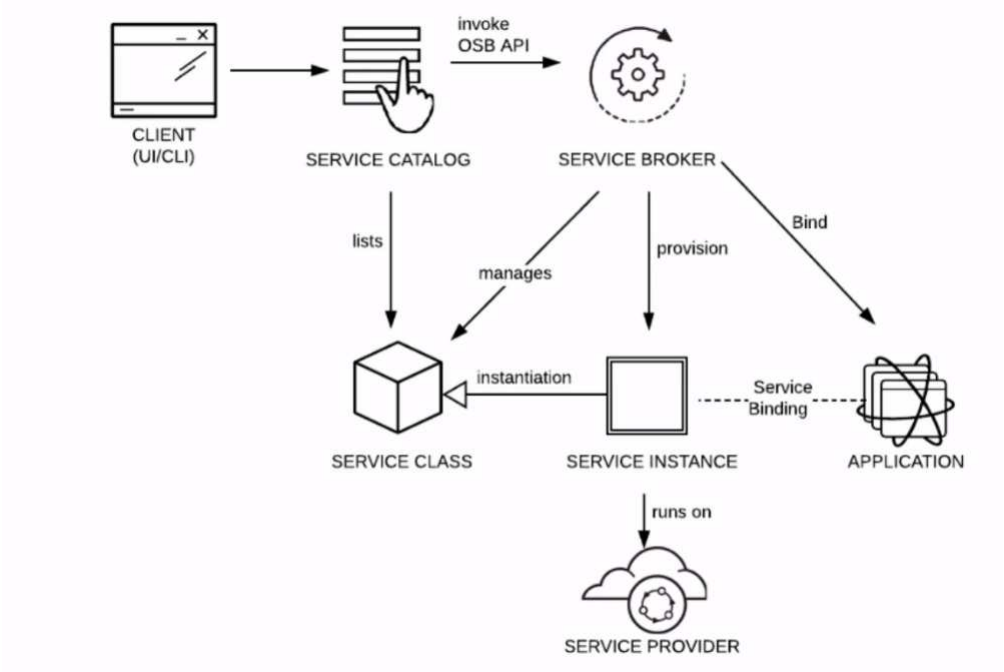


Figure 2.8: Operations of the OSBAPI[26].

implement a service broker in accordance with the OSBAPI specification for ICT services offered at the TU Wien (Section 5.3).

2.5.1 Operations

A service broker can offer many services in its catalog, where a service can basically be anything [MR18]. For example, a service could be a database, a messaging queue, an API gateway, cloud storage, a machine-learning service, etc. A service can be provisioned on-demand for example by launching a new virtual machine or it could be a multitenant service where an existing resource is shared [MR18].

To connect to a provisioned service, a so-called binding is created. Depending on the service, a binding could be a set of credentials plus the IP address and port number of the service, or something else entirely.

Figure 2.8 shows the operations of the OSBAPI. A client retrieves the service catalog from a service broker and gets a list of available service classes / plans. For example, a database could have three plans - small, medium, large - offering different sizes of computing resources. The service broker can provision a service by instantiating a service class that runs on the service provider. By creating a service binding the service can be accessed and used. To use it in an application the binding details can be injected into the runtime environment via environment variables.

[26] https://link.medium.com/iIqQkeQN45

CHAPTER 3

Requirements Engineering

In this chapter we describe how we collected requirements for a maDMP support system. MaDMPs are a relatively new topic that has surfaced in recent years and is not yet part of standard data management practice. Since there are no established use cases for maDMPs, designing a support system is a challenging task. However, through the organization of workshops and consultations [SJMM17][MNWR18], the scientific community identified potential use cases for maDMPs (Section 2.3.1) that form the basis of our work.

When developing our system design, we relied on the ideas and results of the community. Based on the community use cases, we modeled workflows (Section 3.2) and graphical mockups (Section 3.3) that reflect these use cases. The workflows underwent an initial evaluation during a workshop [MCB18] and the mockups were tested and refined with the help of relevant stakeholders at the TU Wien, but also external bodies such as a representative of a major research funding agency in Austria.

3.1 Stakeholders

Based on the community use cases described in Section 2.3.1 and ideas by Miksa et al. [MSMJ19], we can identify six key stakeholders of an institutional, machine-actionable DMP support system. Figure 3.1 depicts these stakeholders and their use cases.

The central use case of the institutional, maDMP support system is the researcher creating a machine-actionable DMP. This use case enables a whole range of other use cases for multiple stakeholders inside and outside the institution.

The next few paragraphs outline the use cases of the stakeholders enabled by a machine-actionable DMP support system. In Section 4.3, the use cases are further elaborated and linked to an institutional RDM policy in order do derive architectural goals and requirements for the system design at an enterprise level.

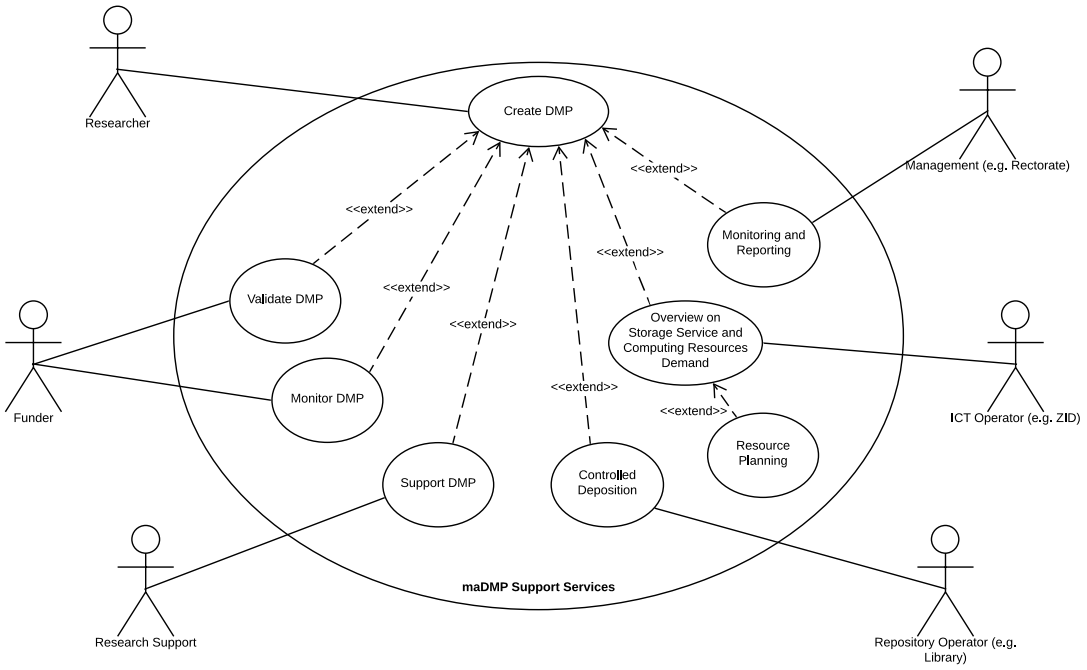


Figure 3.1: Stakeholders and their use cases of an institutional, machine-actionable DMP support system.

Researcher The researcher can get (semi-)automated support in data management planning during the whole lifecycle of a research project. This includes (semi-)automated assistance in pre-filling the DMP with information from related systems, research data specification, cost estimation, storage, service, repository, metadata standard and license selection.

ICT Operator The ICT operator can find out about potential storage service and computing resource demand early in the project and support researchers in making good decisions, to minimize the effort upfront. The ICT operator can get an overview on expected storage service and computing resource demand and better plan future acquisition of resources.

Repository Operator The repository operator can find out about planned deposition of research data and aid researchers in proper data format and metadata standard selection to minimize the effort of data ingest into the repository system.

Research Support The research support can provide assistance in data management planning more efficiently by being integrated into the DMP creation workflow.

Management The institutional management can monitor ongoing data management planning at the institution. Based on the information in the maDMPs created at the institution, the management can generate automated reports about data management practices at the institution and answer questions like:

- What kind and amount of research data is produced?
- Who produces what (e.g. cluster by faculty)?
- How and where is research data preserved?
- Which licenses are used for data sharing?

Funder The funder can get (semi-)automated support in monitoring the DMP throughout the lifecycle of a research project and in validating the compliance of the DMP with its policies in order to make the review process more efficient.

3.2 Automated Workflows

The community use cases described in Section 2.3.1 outline ideas around maDMPs at a coarse-grained level involving all kinds of stakeholders. In order to facilitate the discussion we designed specific workflows in the context of a research institution and their stakeholders. We assume the automated workflows to be embedded in a RDM infrastructure of a research institution including services and people. The proposed workflows are based on and inspired by the community use cases and ideas [SJMM17][MSMJ19][MNWR18] and represent a non-exhaustive list of potential data management planning processes at a research institution.

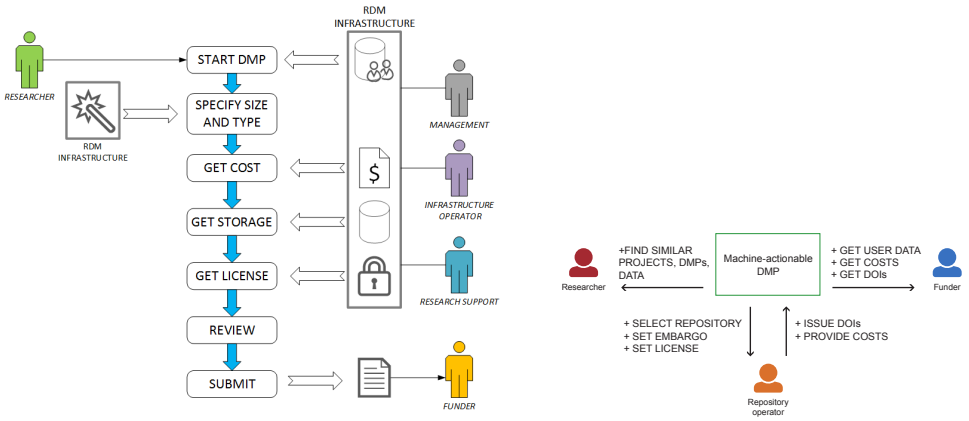
In this section, we describe the automated workflows using the Business Process Model and Notation (BPMN) version 2.0[1] which is a standard notation for describing business processes.

The workflows underwent an initial evaluation in a workshop titled "Domain Specific Extensions for Machine-actionable Data Management Plans" co-organized with the 22nd International Conference on Theory and Practice of Digital Libraries (TPDL 2018) and were published in [MCB18].

3.2.1 Overview

The design of the workflows is framed by the process of creating an initial DMP at an early stage of a project [MNWR18], but also considers data management tasks at later stages, such as when data is ready for submission into a repository system [MSMJ19].

[1] https://www.omg.org/spec/BPMN/2.0/



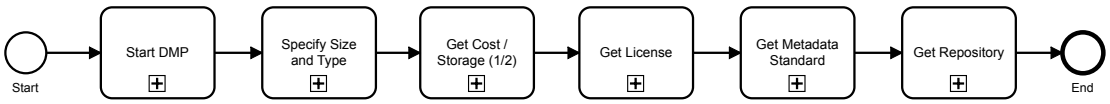
(a) Initial project phase [MNWR18]. (b) Later project phase [MSMJ19].

Figure 3.2: (a) Overview on high-level workflow of creating a DMP at an initial phase of the project. Services and stakeholders of a RDM infrastructure assist the researcher in data management planning tasks, such as research data specification, storage booking, cost estimation or license selection [MNWR18]. (b) Stakeholder services use the maDMP as a medium for information exchange. A repository operator service uses the maDMP to facilitate the submission process of research data into the selected repository system and returns PIDs assigned to the submitted datasets and associated costs. The funder can use the information to check how the DMP was implemented [MSMJ19].

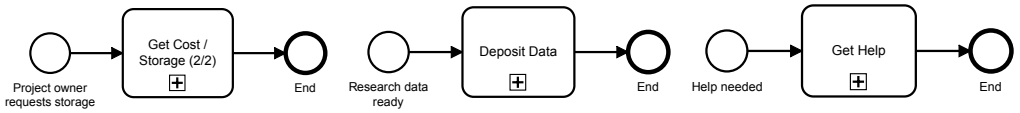
Figure 3.2a depicts a high-level workflow of creating an initial DMP and shows that services and stakeholders of a RDM infrastructure are involved in the process [MNWR18]. In this high-level workflow, the researcher starts a DMP and administrative information like researcher affiliation is fetched from related information systems to pre-fill the DMP. Next, the researcher estimates the expected size and type of the research data and gets assistance in storage booking, cost estimation and license selection.

Figure 3.2b shows potential interactions between stakeholder services using a maDMP as a medium for information exchange at a later stage in the project [MSMJ19]. Information contained in the maDMP like specified license or embargo period can be used to facilitate the submission process of research data to the selected repository by automatically assigning these fields in the repository system. A repository operator service can return a list of PIDs pointing to the submitted datasets and provide associated cost to update the maDMP. A funder service can use the information to check how the DMP was implemented.

The high-level workflows described so far remain vague, and it is not clear what exactly should be done at each step. Hence, we modeled nine sub-workflows using the BPMN. Figure 3.3 gives an overview on the high-level BPMN workflows. In Sections 3.2.2 to 3.2.9 we describe these sub-workflows, which form the foundation for our system architecture design, in more detail.



(a) Workflow of creating an initial DMP.



(b) Provision storage. (c) Deposit data. (d) Get help.

Figure 3.3: Overview of the automated workflows using BPMN.

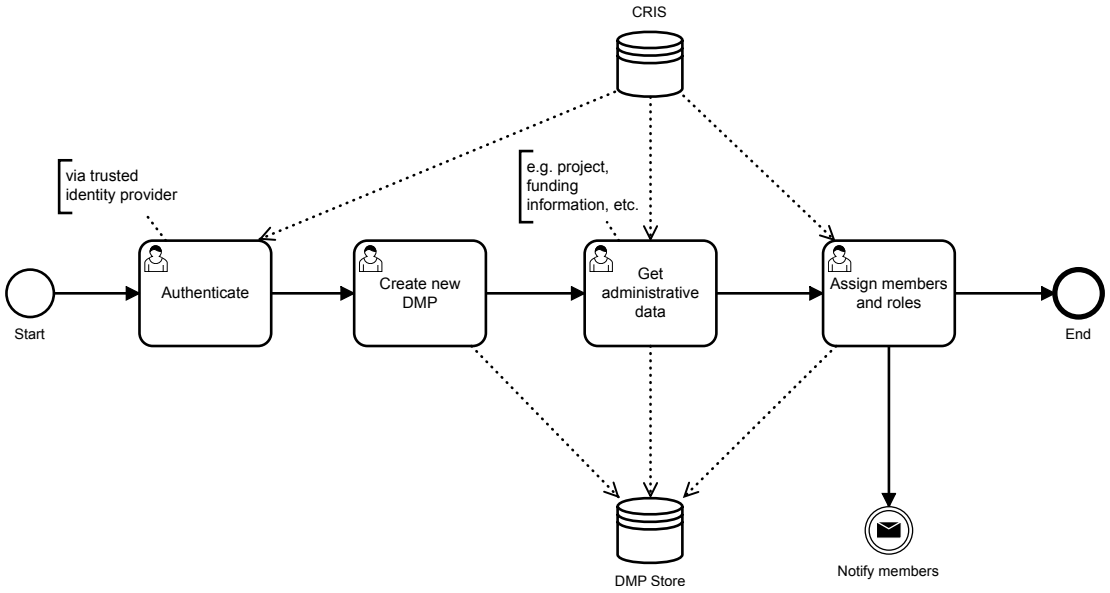


Figure 3.4: *Start DMP* workflow - creation of an initial DMP.

3.2.2 Start DMP

The *Start DMP* workflow, as depicted in Figure 3.4, is all about creating an initial DMP.

1. **Authenticate**: The researcher authenticates through a trusted identity provider. The researcher's identity can be authenticated through the institutional authentication mechanism or through external systems such as the Open Researcher and Contributor ID (ORCID)[2] registry.

2. **Create new DMP**: The researcher creates a new DMP which gets persisted to the DMP Store database.

[2]https://orcid.org/

3. **Get administrative data**: Administrative information associated with the signed-in institutional CRIS or ORCID account is retrieved via their APIs and imported into the DMP. In addition to personal data of the researcher such as person ID, ORCID, email or affiliation, information about work and projects can also be imported. If a researcher wants to create a DMP for a work or project that is already registered in the institutional research project database or in the ORCID record, a list of works or projects can be displayed for selection and information can then be imported into the DMP. Project funding information such as funder ID or project grant number can also be imported from suitable information systems. The Crossref Funder Registry[3] provides a curated, worldwide list of research funding organizations which allows to uniquely identify and reference funders, e.g. the Austrian Science Fund (FWF)[4].

4. **Assign members and roles**: The researcher can add collaborators to the DMP, e.g. other researchers involved in the project by searching through the APIs of the CRIS or ORCID registry and importing person details. Collaborators can be assigned with roles and responsibilities for the data management. When the research data management team is set-up, a notification informing each member about the DMP and their roles/responsibilities is sent. Each member uniquely identified by their institutional identifier, ORCID or email is granted access to the DMP.

3.2.3 Specify Size and Type

The *Specify Size and Type* workflow, illustrated in Figure 3.5, deals with the specification of research data that will be (re-)used and generated during the project.

1. **Cite data**: If existing, citable datasets are being reused, their unique and resolvable PIDs like DataCite[5] DOIs can be entered and associated metadata be retrieved and imported into the DMP. Many repositories support the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) standard, other repositories like GitHub provide a REST-API to collect metadata. Citing reused datasets in a DMP enables the data discovery use case for datasets referenced in different DMPs (Section 2.3.1). Retrieved metadata such as the license of the reused dataset clarify the terms of use [Bal14] and can avoid pitfalls already in the planning phase.

2. **Specify output data**: The researcher can specify which data will be generated during the project. The researcher can specify the expected types, file formats and volumes of research data manually or get automated support by uploading datasets which are analyzed (file format, size, other metadata) by a file identification and characterization tool. If entire datasets are already known or available, they can be

[3] https://www.crossref.org/services/funder-registry/
[4] http://dx.doi.org/10.13039/501100002428
[5] https://datacite.org/dois.html

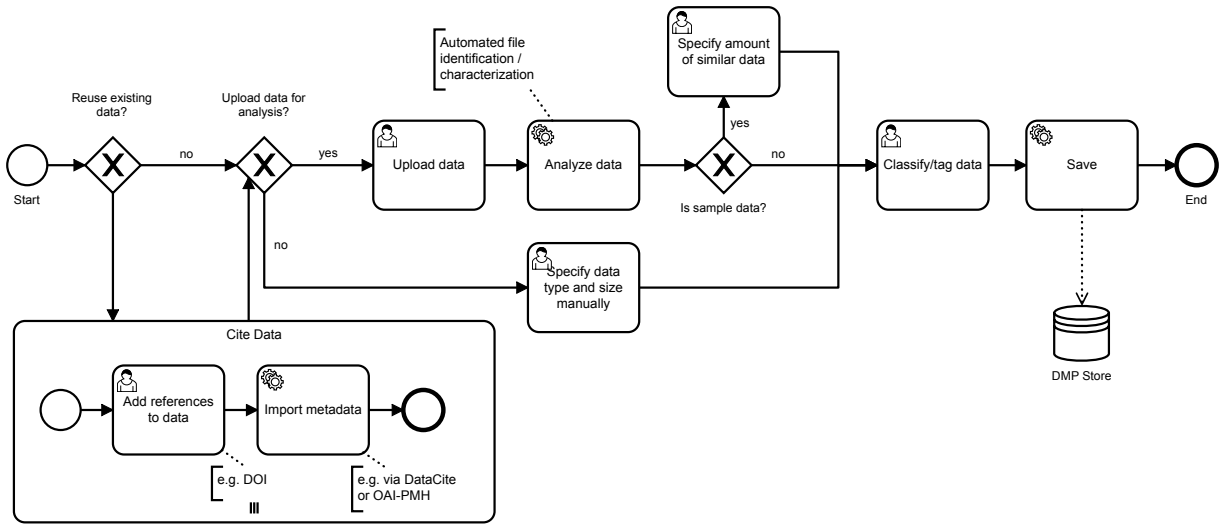


Figure 3.5: Specify size and type of research data.

uploaded for analysis. Alternatively, sample data can be uploaded and analyzed, the expected number of similar files can be specified, and an estimate of the total size can be calculated.

3. **Classify/tag data**: The researcher can classify/tag the data, e.g. by assigning labels.

4. **Save**: The workflow concludes with storing the details of the research data specification to the DMP.

3.2.4 Get Cost and Storage

The *Get Cost and Storage* workflow deals with the discovery, selection, configuration, cost estimation and provisioning of various kinds of storage and other ICT services used for managing research data during the project (active data). The workflow is divided into two phases. In the first phase (Figure 3.6) the researcher discovers, selects and configures required services offered by the ICT operator and gets a cost estimation. In the second phase (Figure 3.7), when the ICT services are requested by the project owner, the services are actually booked and provisioned. If expensive computing resources are requested, the project funding status can also be checked to avoid stranded costs.

By splitting up the service provisioning process into two phases, we can achieve a realistic cost estimation of ICT services required for data management at an early stage and provide the ICT operator with information about upcoming service demands.

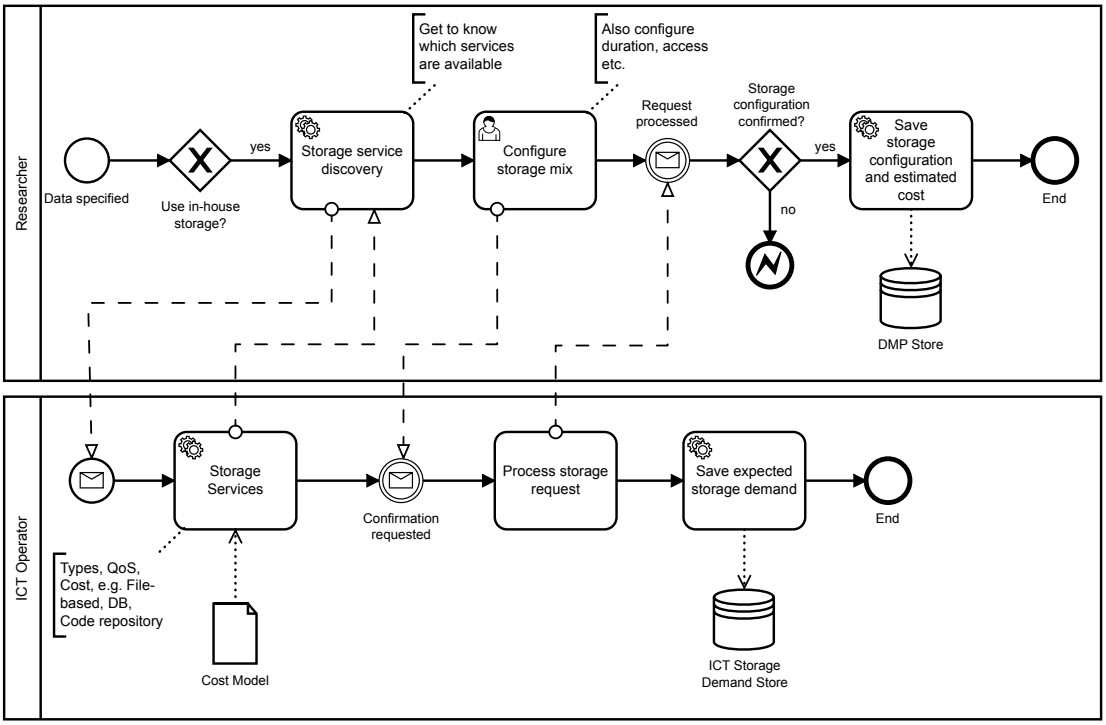


Figure 3.6: *Get Cost and Storage* workflow (1/2), storage mix configuration and cost estimation.

Storage Configuration and Cost Estimation

The workflow shown in Figure 3.6, assumes that the type and size of the research data is specified (see Section 3.2.3).

1. **Storage discovery**: If in-house storage services are to be used, a storage discovery service helps the researcher find out which services are available and seem fit for the type of data at hand. The storage discovery service communicates with the storage services offered by the ICT operator. Information about the types of available storage services (e.g. file-based, database, code-repository) and associated QoS attributes, such as access speed, availability, backup type, etc. is provided. Based on an internal cost model and the desired configuration, the associated costs can be provided.

2. **Configure storage mix**: Once the researcher knows about the available storage services and costs, he or she can configure a mix of required storage services and provide further information, like who should have access and the period the services are needed. A request containing the desired mix of storage services is sent to the ICT operator for further processing.

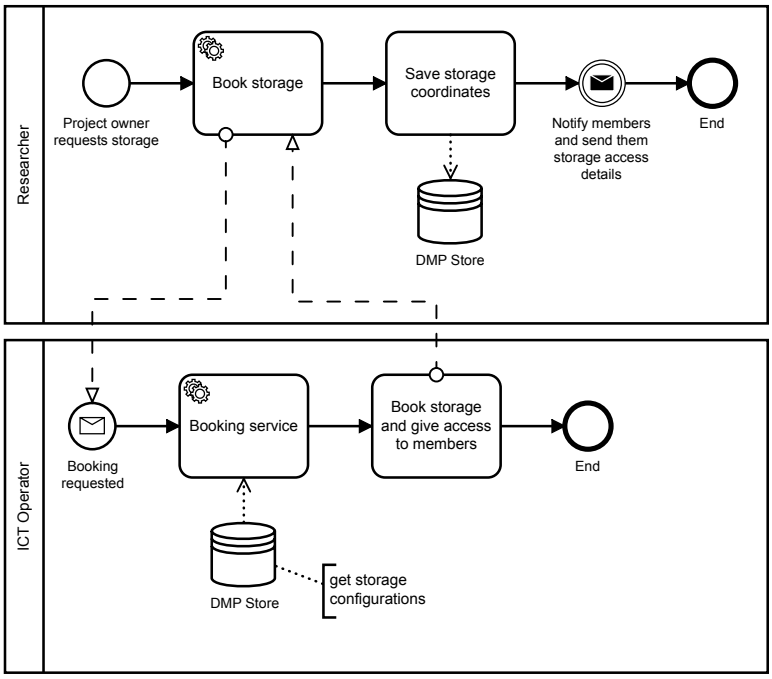


Figure 3.7: *Get Cost and Storage* workflow (2/2), storage provisioning after being requested by the project owner.

3. **Save storage configuration and estimated cost**: If the requested mix of storage services is confirmed, the details and associated costs are stored in the DMP. On the ICT operator side, the storage request is stored and the expected computing resources demand is known for the near future, but no service booking/provisioning is done yet.

Storage Provisioning

The *Storage Provisioning* workflow, depicted in Figure 3.7, is triggered when the storage is requested by the project owner.

1. **Book storage**: A book storage task requests the ICT operator to book the previously configured mix of storage services. A booking service retrieves the details of the service configurations and triggers the actual booking of the services and starts the associated service provisioning processes, such as setting-up a code-repository, spinning up a virtual machine, mounting a volume, etc.
2. **Give access**: As soon as the storage services are ready, team members are given access and the coordinates of the services (e.g. access links) are stored to the DMP.
3. **Notification**: All team members receive a notification of the availability of the storage services, including their access details.

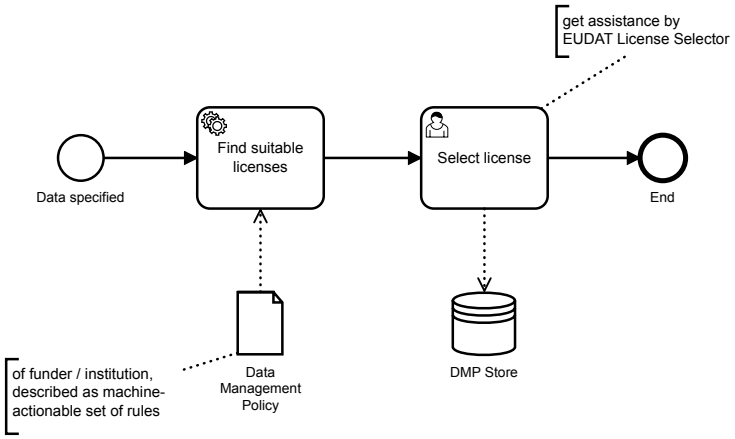


Figure 3.8: Workflow of license selection for the publication of research data.

3.2.5 Get License

The *Get License* workflow (Figure 3.8) assists the researcher in selecting suitable licenses for the publication of research data.

1. **Find license base on policy**: Based on the policy of the funder or the institution, a license can automatically be selected under which the research data should be published. For example, an institutional research data policy could recommend publishing source code under the GPL license and other datasets under the CC-BY license [TU 18]. Miksa et al. [MSMJ19] suggest to make data policies also machine-actionable, e.g. by transforming parts of a policy into a machine-actionable set of rules and include them into policy documents. A license selection service can then consume the machine-actionable policies and help the researcher in license selection.

2. **Select license**: If the data policy is non-restrictive, a license selection wizard such as the EUDAT License Selector[6] can be integrated to guide the researcher through the decision process of selecting suitable licenses [MSMJ19]. After the license is selected, the decision is stored to the DMP.

3.2.6 Get Metadata Standard

The *Get Metadata Standard* workflow, depicted in Figure 3.9, supports the researcher in selecting suitable metadata standards/formats to describe the research data for better reusability.

1. **Metadata standards discovery**: The researcher consults a metadata standard discovery service that allows to browse through a list of standards and apply search

[6] https://eudat.eu/services/userdoc/license-selector

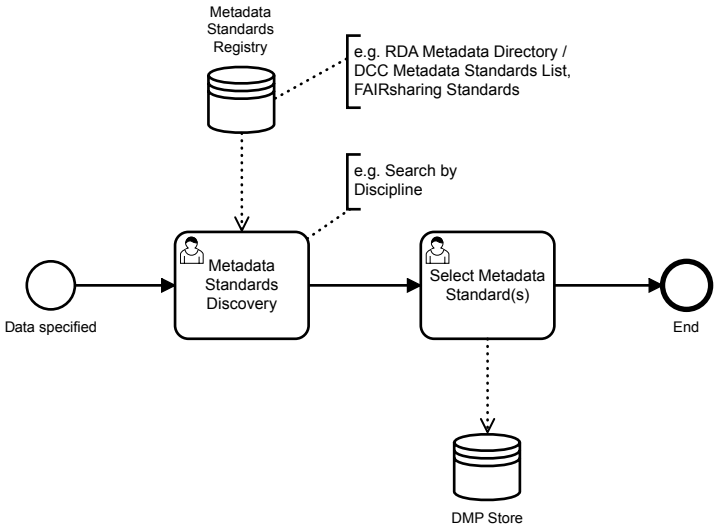


Figure 3.9: Workflow of selecting metadata standards to describe research data for better reusability.

filters, e.g. filter by discipline or type of data. The service can be backed-up by a metadata standards registry which maintains a curated list of metadata standards such as the RDA Metadata Directory[7] based on the standards list from the DCC[8] or the standards list from FAIRsharing[9]. The guidelines of the EC on FAIR data management in the Horizon 2020 programme recommends to consult the RDA Metadata Directory to find discipline-specific standards and tools [Com16].

2. **Metadata standards selection**: When suitable metadata standards are found and selected, the selection gets persisted to the DMP.

This workflow is inspired by the functionality implemented in the DMPRoadmap[10] platform, which integrates the RDA Metadata Directory into the tool.

3.2.7 Get Repository

Research data repositories play an essential role in the dissemination and preservation of FAIR data [HJC+18]. There exist thousands of disciplinary and generalist repositories for research data. Hence, finding and selecting suitable repository systems can be challenging. The *Get Repository* workflow (Figure 3.10) deals with automated support for the selection of research data repository systems for the sharing and long-term preservation of research data.

[7] http://rd-alliance.github.io/metadata-directory/
[8] http://www.dcc.ac.uk/drupal/resources/metadata-standards
[9] https://fairsharing.org/standards/
[10] https://github.com/DMPRoadmap/roadmap/releases/tag/v2.1.3

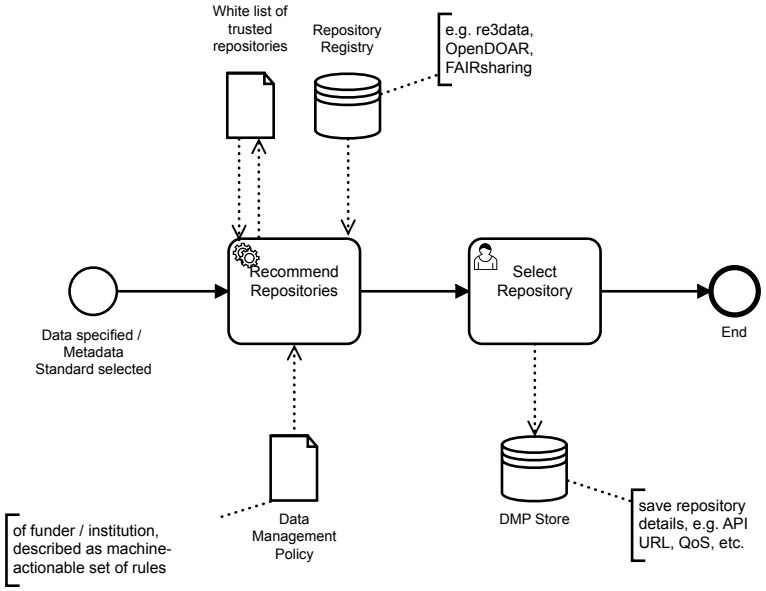


Figure 3.10: Workflow of selecting repository systems for the sharing and long-term preservation of research data.

1. **Recommend repositories**: The researcher can consult a repository recommendation service to find suitable repository systems for the sharing and long-term preservation of research data. The repository recommendations can be based on information provided in the maDMP, as well as on the funder's/institution's repository selection criteria such as the support of FAIR data principles [WDA+16][HJC+18] or the repository's trustworthiness [Sci18a]. Therefore, the repository recommendation can have input parameters such as the field of study, type and format of research data, used metadata standards, desired PID systems supported by the repositories (e.g. DOI), repository certification (e.g. Core Trust Seal (CTS)), or geographic location of the repository. Similar as described in the *Get License* (Section 3.2.5) workflow, funder's/institution's repository selection criteria could be described as machine-actionable data policy elements and be consumed by the repository recommendation service. The repository recommender service can be backed-up with curated and programmatically accessible lists of research data repositories available from the re3data[11], the Directory of Open Access Repositories (OpenDOAR)[12] or FAIRsharing[13]. Re3data and OpenDOAR provide extensive metadata for each indexed repository system that describe repository attributes and enable filters to be applied to attributes to find matching repository systems. FAIRsharing maintains collections of repository systems that are recommended by

[11] https://www.re3data.org/
[12] https://v2.sherpa.ac.uk/opendoar/
[13] https://fairsharing.org/databases/

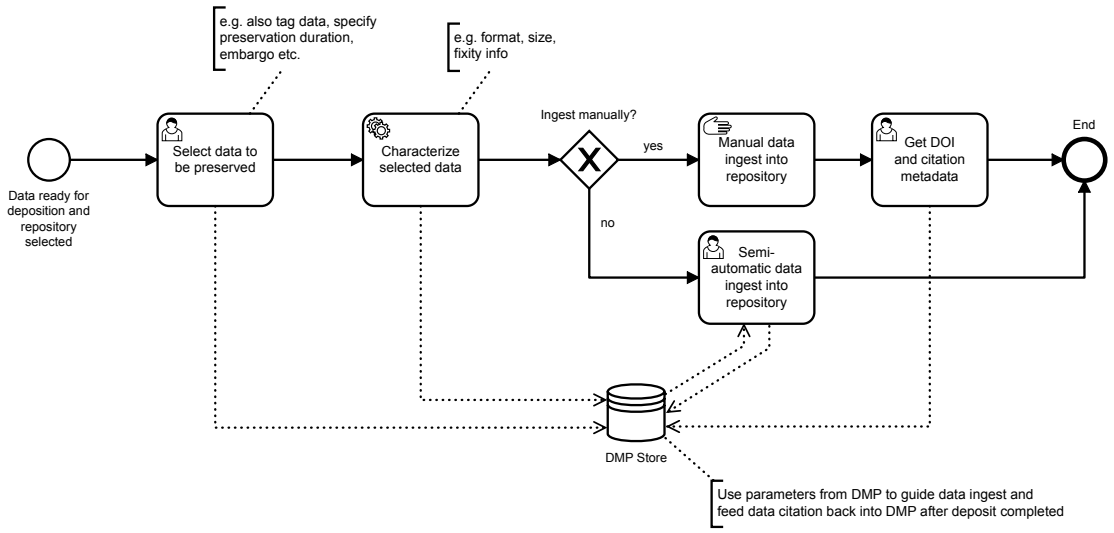


Figure 3.11: Workflow of depositing research data in a repository system.

journals[14] for the publication of research data that underlie journal publications. The repository recommendation service could implement repository ranking schemes based on institutional needs and policies or whitelist institutional repository systems to advertise their use. While the repository recommendation described so far is a content-based approach, repository recommendation can also be the result of collaborative-filtering, taking into account the repository selections of other researchers, e.g. with similar data or field of study.

2. **Repository selection**: When suitable repositories are found and selected, metadata related to the repository systems, such as API endpoints (e.g. OAI-PMH) or QoS descriptions are saved to the DMP.

3.2.8 Deposit Data

The *Deposit Data* workflow (Figure 3.11) models the process of a researcher depositing research data in a repository system and updating the maDMP automatically.

1. **Data selection**: The researcher can select which datasets should be deposited in repository systems for sharing and long-term preservation [DCC14]. For that purpose the researcher can be presented with a structured list of files located in the active data storage to select from, or select files from the local file system. The researcher then can provide metadata (e.g. tags, embargo period, preservation duration, license) for the datasets if not already specified in the maDMP. A file characterization tool can analyze the datasets and provide additional metadata such as size, file formats and fixity information.

[14] https://fairsharing.org/recommendations/

2. **Data submission**: The researcher can choose whether the research data should be submitted manually to the repository system or semi-automatically if the repository supports automated data ingest via an API. A standard way of depositing digital objects into a repository is the Simple Web-service Offering Repository Deposit (SWORD) protocol[15]. Common repositories supporting SWORD are Dataverse[16], DSpace[17], or EPrints[18]. The metadata provided in the maDMP such as embargo periods and licenses can be used to automatically set this information as metadata in the repository system [MNWR18][MSMJ19]. SWORD specifies a default format for metadata deposit using a JSON-LD serialization with Dublin Core vocabulary elements from dc[19] and dcterms[20], but also supports arbitrary metadata schemes and serialization formats. However, a standard metadata schema could be specified that describes common parameters for data deposits in repositories and is compatible with standard maDMPs to improve the interoperability between maDMPs and a wide range of repository systems. Repositories that support a common metadata schema for data deposits can improve the user experience, since information that is already specified in the maDMP would not have to be entered again during the data submission process.

3. **Get DOI and citation metadata**: After the data deposit into the repository system completed, the locations of the datasets (e.g. DOIs) and citation metadata provided by the repository can be used to update the maDMP.

3.2.9 Get Help

Figure 3.12 depicts the *Get Help* workflow which illustrates the integration of research support into data management planning.

1. **Request help**: If a researcher needs help/advice with data management planning, he or she can create a help request which will be sent to a research support help desk.

2. **Notification**: Employees of the help desk get notified, can process the help request and provide feedback. The help desk could use some kind of ticketing system and get access to the DMP for the time of support.

3.3 Graphical Mockups

In Section 3.2 we described automated workflows for data management planning using BPMN and textual descriptions. BPMN is a notation used to facilitate discussions

[15] https://swordapp.github.io/swordv3/swordv3.html
[16] SWORDv2 http://guides.dataverse.org/en/latest/api/sword.html
[17] SWORDv2 https://wiki.lyrasis.org/display/DSDOC6x/SWORDv2+Server
[18] SWORDv2 https://wiki.eprints.org/w/SWORD_2.0
[19] http://purl.org/dc/elements/1.1/
[20] http://purl.org/dc/terms/

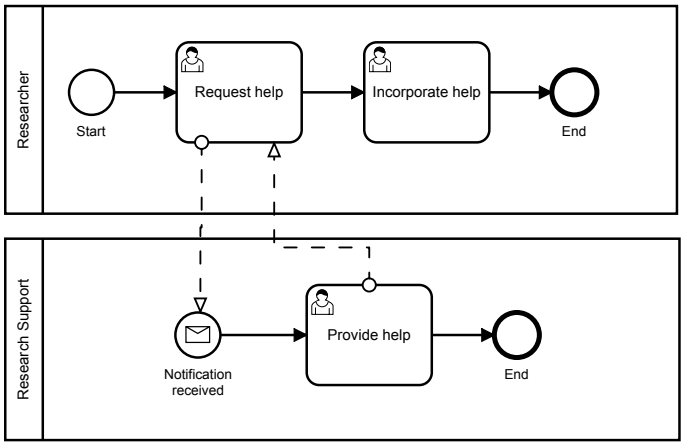


Figure 3.12: *Get Help* workflow - getting help in data management planning from research support.

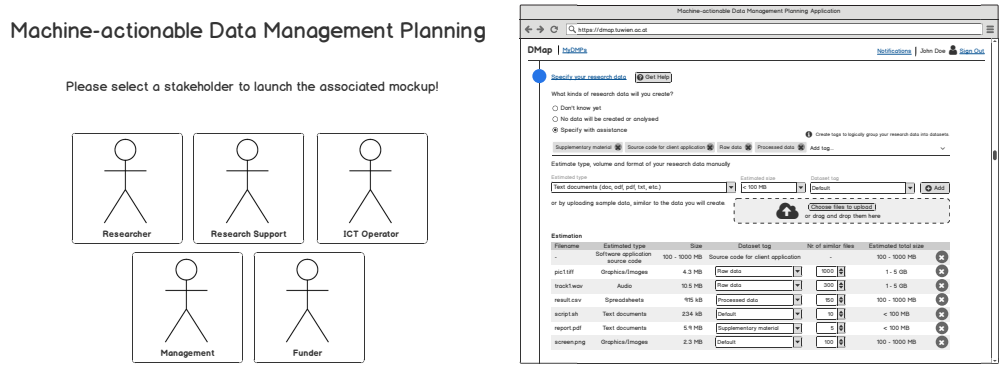
about business processes between non-technical and technical people by using a common language. However, from a UI design perspective it is not straight-forward to derive requirements for an UI from the BPMNs. UI mockups also present a quick opportunity to test concepts with stakeholders and receive valuable feedback on the further design. Therefore, we developed interactive, graphical UI mockups [Obl20a] using Balsamiq[21] for five different stakeholder groups.

1. Researcher
2. Research Support
3. ICT Operator
4. Management
5. Funder

All of these stakeholders are involved in the data management planning workflows and related use cases. The graphical UI mockups are composed of about 60 wireframes. The wireframes are linked with each other and contain interactive, clickable elements. We published the graphical mockups on GitHub pages[22] as pdf and with a link to the interactive version hosted at Balsamiq Cloud. Visitors of the Balsamiq Cloud could provide comments with markers directly on the wireframes. Figure 3.13 shows the UI mockups start screen and a sample wireframe designed for a researcher.

Additionally, we met in person with representatives from the stakeholder groups located in Austria and presented the mockups and documented their feedback. With researcher,

[21] https://balsamiq.com/
[22] https://oblassers.github.io/dmap-mockups/



(a) Mockups start screen. (b) Wireframe for specifying research data.

Figure 3.13: Graphical UI mockups were published as pdf and interactive version on GitHub pages. Visitors could provide feedback directly on the wireframes. Additionally, feedback was collected personally from representatives of stakeholder groups [Obl20a].

research support, and funder as key players in data management planning, we met with representatives of these groups of stakeholders as illustrated in Table 3.1. Most of the stakeholder representatives are from departments of the TU Wien. The European and International Research Support[23] (EIRS) at the TU Wien supports researchers in preparing and carrying out European and international research projects. The TU Wien Center for Research Data Management[24] assists researchers in managing their research data over the entire data lifecycle and is the main contact point for DMP-related questions. Depending on the field of study the type of data and common data practices can vary dramatically. Therefore, we collected feedback from researchers of two different departments, namely the Research Group for Sustainable Technologies and the Research Unit of Remote Sensing at the TU Wien. The FWF is a major funder of scientific research in Austria and requires the submission of a DMP for research projects granted from January 2019. We could collect feedback from the main contact person for DMPs at FWF. We also received feedback from the DMPTool product manager who is an established maDMP expert and co-author of the maDMP whitepaper [SJMM17].

The feedback collected from the stakeholders contributed to the improvement of the graphical mockups. In this work, we went through several iterations of incorporating feedback, with v1.2 [Obl20a] being the latest version of the mockups.

[23] https://www.tuwien.at/en/tu-wien/organisation/service-providers/rti-support/international-research-support/

[24] https://www.tuwien.at/en/research/rti-support/research-data/

Table 3.1: Stakeholder groups and their representatives from departments of organizations to whom the graphical UI mockups were presented and gave feedback.

Stakeholder Group	Representative from	Organization
Research Support	European and International Research Support	TU Wien
	Center for Research Data Management	
Researcher	Research Group for Sustainable Technologies and Process Simulation	
	Research Unit of Remote Sensing	
Funder	Strategy - Policy, Evaluation, Analysis	FWF

3.4 Summary

The scientific community identified many use cases for maDMPs, as described in Section 2.3.1. These use cases describe a vision of a more integrated and automated data management with potential benefits for many stakeholders. The RDA DMP Common Standards working group collected requirements for a machine-actionable DMP data model (Section 2.3.1). However, the community use cases are described at a coarse-grained level and the requirements engineering conducted by the RDA working group aimed on the design of a machine-actionable DMP data model. For this reason, it is not clear what the surrounding processes should look like. We therefore created nine specific workflows of machine-actionable data management planning based on the community ideas using the BPMN. The BPMN workflows can be understood by technical and non-technical people and facilitate the discussion about the design of the processes. The BPMN processes underwent an initial evaluation [MCB18] and form the basis for further discussion.

Based on the BPMN processes, we created graphical mockups for five different stakeholder groups and collected feedback from group representatives to derive system requirements and to improve the mockups accordingly.

We are aware that more requirements engineering needs to be conducted. However, with our exploratory work we hope to advance the discussion and contribute to a better understanding of maDMPs and the surrounding processes.

CHAPTER 4

Enterprise Architecture

In this chapter we introduce an architecture for machine-actionable data management planning in the context of a research institution and its infrastructure. A research institution or university with its various stakeholders, systems and services can be viewed as an enterprise. We therefore use a software engineering technique called EA Modeling to describe the architecture from various perspectives, highlighting different aspects.

The workflows described in Chapter 3 define the business processes of our EA which are essential for the system design.

4.1 Introduction

According to Giachetti [Gia10], an enterprise is a complex socio-technical system that involves people, information and technology interacting with each other and their environment to pursue a common mission.

Giachetti describes the methodology of designing an enterprise as developing a "best" system solution that satisfies the requirements of all stakeholders. He states that this is an iterative process and takes multiple iterations of understanding stakeholder perspectives and needs, analyzing, designing and evaluating before the final enterprise design is found.

The initial EA described in this chapter forms the basis for finding the "best" system solution.

4.1.1 Purpose

The proposed EA can help in establishing machine-actionable data management plans and associated processes at research institutions. The architecture describes how machine-actionable data management planning processes can be implemented within the scope of

a research institution in order to make data management planning an integral part of research, beneficial for all stakeholders.

The purpose of the architecture description[1] is to give a comprehensive overview of the system by using different architectural views to highlight different system aspects, capture significant architectural decisions and communicate them to various stakeholders. According to The Open Group (TOG) [Gro17] the architectural views serve the purposes of designing, deciding and informing.

- Designing: support architects in the design process
- Deciding: assist decision-makers by providing insights into cross-domain architecture relationships
- Informing: inform stakeholders about the architecture to obtain commitment and convince opposers

4.1.2 Scope

The EA describes a machine-actionable data management planning support system embedded into an institutional RDM infrastructure. The design is based on the use cases and workflows described in Chapter 3. It describes which services are needed and how they can be integrated with existing institutional services such as research information systems or services from ICT providers.

The proposed Enterprise Architecture describes the intended system from a high-level point of view and does not make a statement on any implementation details like specific frameworks, programming languages or tools.

4.1.3 Overview

This chapter is following the structure of the Software Architecture Document (SAD) template[2] of the Rational Unified Process (RUP). The sections of this chapter are organized as follows.

- Section 4.2 outlines how the architecture is represented, which architectural views and notations are chosen for which purpose.
- Section 4.3 describes the architectural goals and derived requirements and constraints that may apply.
- Section 4.4 provides an overview on the structure and composition of business processes needed to realize previously described use cases.

[1] https://sce.uhcl.edu/helm/RationalUnifiedProcess/process/artifact/ar_sadoc.htm
[2] https://sce.uhcl.edu/helm/RationalUnifiedProcess/webtmpl/templates/a_and_d/rup_sad.htm

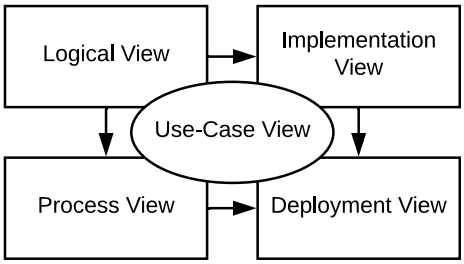


Figure 4.1: Kruchten's 4+1 architectural view model [Kru95].

- Section 4.5 presents an introductory and layered view to give an overview and communicate the relationships between business, application and technology layers of the architecture.
- Section 4.6 describes the runtime behaviour of selected use cases as sequence diagrams.
- Section 4.7 gives an overview on the application structure, its components and the information flow between components.
- Section 4.8 describes how hardware and software infrastructure support the deployment of application components and discusses two potential service integration options.

4.2 Architectural Representation

The architecture description and representation is based on Kruchten's 4+1 architectural view model [Kru95], depicted in Figure 4.1. It uses multiple, concurrent views to describe the architecture from the viewpoints of different stakeholders, such as end-users, developers, system engineers or project managers, addressing their concerns.

Most of the views for this architecture are created with the ArchiMate Enterprise Architecture modelling language [Gro17]. ArchiMate is an open standard, hosted by TOG and compliant with The Open Group Architecture Framework (TOGAF) [Gro18]. ArchiMate supports the paradigm of viewpoint-oriented architecture description and allows to create models enabling the description, analysis and communication of different stakeholder concerns. Each viewpoint serves purposes such as designing a potential architecture, informing the stakeholders or aid in decision making. Views are interrelated and should not be viewed in isolation, they always give just a partial, incomplete picture of the system. The combination of views, with their inter-dependencies allows to describe and communicate aspects of the architecture [Gro17].

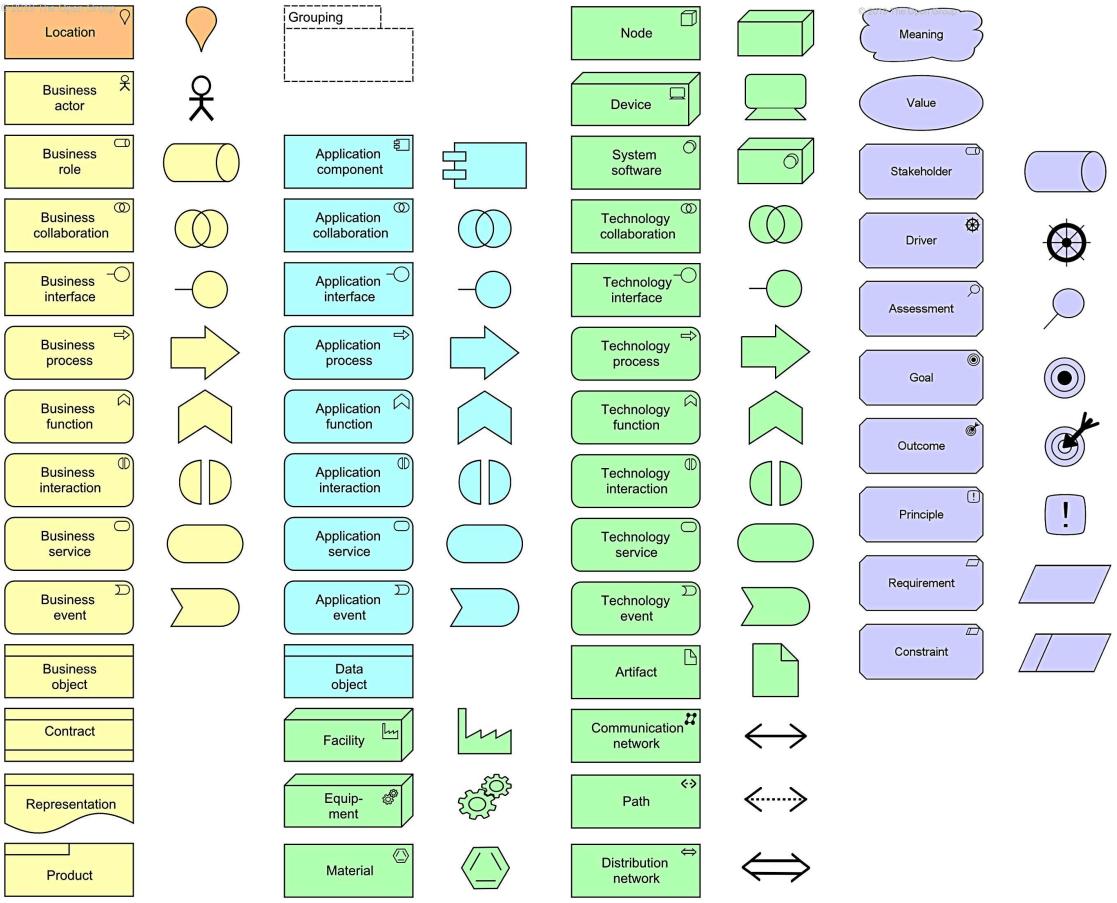


Figure 4.2: Elements in the ArchiMate language notation [Gro17].

4.2.1 Notation

ArchiMate follows a service-oriented approach to differentiate and relate layers of the EA. For the modelling of this architecture, the following color-coded layers of the ArchiMate notation are used.

- Business layer (yellow): describing stakeholders, roles, business services, processes, etc.
- Application layer (blue): describing application services, components, interfaces, etc.
- Technology layer (green): describing physical and technological elements, such as computing nodes, networks or system software
- Motivation layer (violet): describing stakeholders and their motivations, e.g. their drivers for change, goals, related requirements and constraints

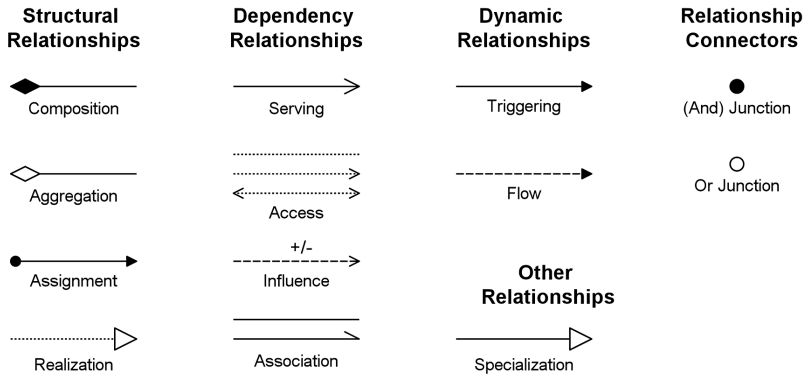


Figure 4.3: Relationships in the ArchiMate language notation [Gro17].

Table 4.1: Mapping from Kruchten's 4+1 views to views used to describe this architecture.

Kruchten's 4+1 views	Views used in this architecture description	
	View	Notation
Use-Case view	Business process view	ArchiMate
Logical View	Introductory view	ArchiMate
	Layered view	ArchiMate
Process view	Sequence diagrams	UML
Implementation view	Application structure, co-operation view	ArchiMate
Deployment view	Infrastructure usage view	ArchiMate
-	Stakeholder view	ArchiMate
	Goal realization view	ArchiMate

Figure 4.2 shows the elements of the color-coded layers in ArchiMate notation [Gro17]. To describe the relationships between these elements, the ArchiMate notation shown in Figure 4.3 is used [Gro17].

4.2.2 Views

The RUP's SAD organizes the architecture description based on Kruchten's 4+1 views, which are represented by Unified Modeling Language (UML) diagrams. However, to describe the architecture on an enterprise level we use the ArchiMate notation. Therefore, we apply a mapping from Kruchten's 4+1 views to ArchiMate views, as described by TOG [Arm13]. We use UML sequence diagrams only to represent Kruchten's process view. We use ArchiMate's motivation extension to describe stakeholder's motivation and goals. Table 4.1 shows the mapping and gives an overview of the used architecture views and their notations. Table 4.2 shows the audiences and their concerns that the selected views should address [Gro17].

Table 4.2: Views used in this architecture description and their intended audiences and concerns they address [Gro17].

View	Audience	Concerns
Business process view	Process and domain architects, operational managers	Structure of business processes, consistency and completeness, responsibilities
Introductory view	Enterprise architects, managers	Make design choices visible, convince stakeholders
Layered view	Enterprise, process, application, infrastructure, and domain architects	Consistency, reduction of complexity, impact of change, flexibility
Sequence diagrams	Integrators	System runtime behaviour, non-functional requirements
Application structure and co-operation view	Enterprise, process, application, domain architects	Application structure, relationships and dependencies between applications, orchestration/choreography of services, consistency, completeness, reduction of complexity
Infrastructure usage view	Application, infrastructure architects, operational managers	Dependencies, performance, scalability
Stakeholder view	Business managers, enterprise and ICT architects, business analysts, requirements managers	Architecture mission and strategy, motivation
Goal realization view	Business managers, enterprise and ICT architects, business analysts, requirements managers	Architecture mission, strategy and tactics, motivation

4.3 Architectural Goals

In this section we describe the goals and associated requirements and constraints that have a significant impact on the design of the EA. For the description, we use ArchiMate's stakeholder and goal realization views. The stakeholder view shows the stakeholders associated with their internal and external drivers for change. We assess each driver for change in strengths, weaknesses, opportunities and threats (SWOT). This assessment helps us to identify goals and to derive associated requirements and constraints for the achievement of these goals. The assessment is based on the use cases described in Chapter 3.

The stakeholder view depicted in Figure 4.4 shows all the relevant stakeholders of the system, their drivers for change, the assessment and the relation to the identified goals. We assume that the institutional policy for research data management is the main driver for change. The RDM policy regulates the handling of research data at the institution and defines responsibilities of the researchers and the institution. For example, the RDM policy of the TU Wien [TU 18] stipulates that researchers are responsible for the creation, maintenance and implementation of DMPs for research projects, while the institution is responsible for providing services and infrastructure to support the researchers.

The institutional research data management policy can be divided into the following sub-drivers for change. However, funders have their own policies for managing research data that may have precedence over the institutional RDM policy.

- Data management planning
- Active data management - management of research data during the research project
- Long-term preservation of research data
- Support researchers in data management

Note that active data management and long-term preservation of research data may be achieved through an integrated process where data is collected, processed, and archived in a way that is transparent to researchers.

4.3.1 SWOT Analysis

For each of the identified drivers for change we assessed Strengths (S), Weaknesses (W), Opportunities (O) and Threats (T). Figure 4.4 illustrates the results of our assessment.

A current weakness is the lack of automated support for DMPs within research institutions. Therefore, we can identify the goal of providing a semi-automatic support system for DMPs. Institutional stakeholders such as the ICT operator, the library or research support can draw synergies from such a system by integrating them into the process of data management planning.

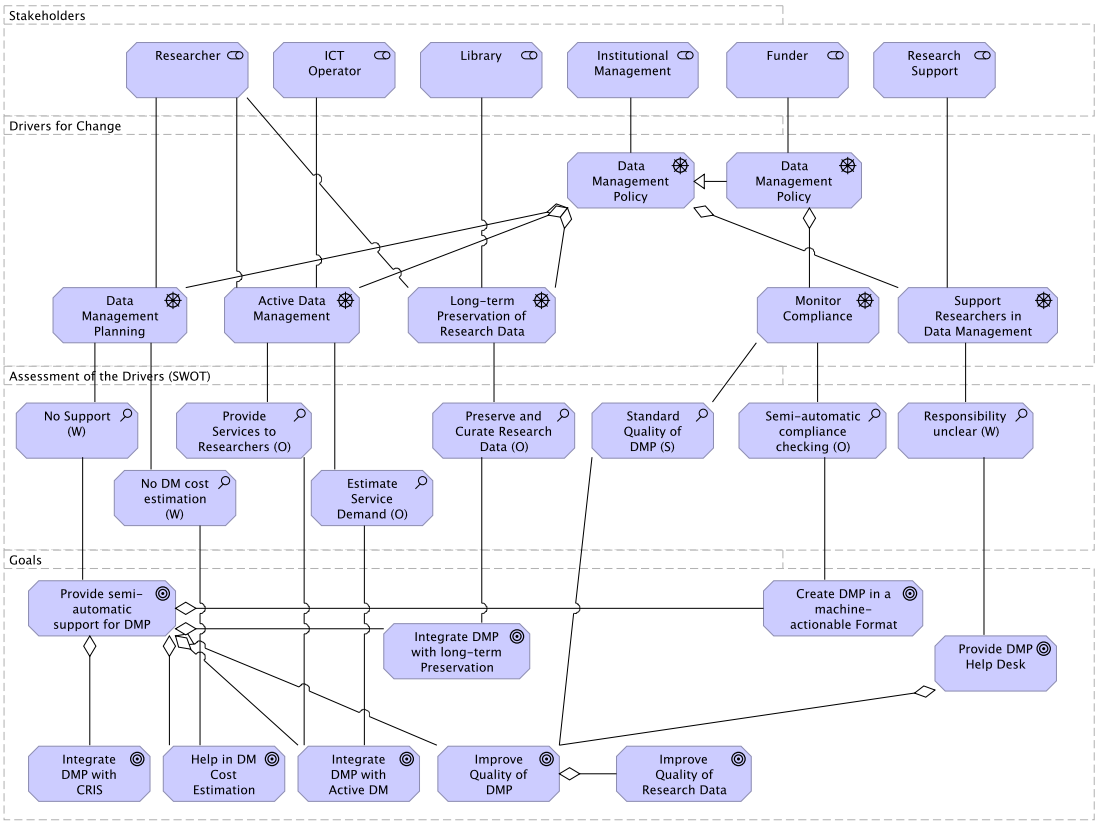


Figure 4.4: Stakeholder view illustrating their drivers for change, the associated assessments of strengths, weaknesses, opportunities and threats (SWOT) and the resulting goals.

ICT services for active data management can be integrated with DMPs to increase their utilization and allow better capacity planning based on estimated service demands described in DMPs.

The operator of a repository for long-term preservation of research data, can benefit from a DMP integration by receiving timely information about research data and assisting with curation and preservation tasks to ensure a smooth transition to the repository.

Research support can benefit from a DMP integration by having tool support to provide assistance and help improve the quality of DMPs. A possible weakness is that responsibilities of the research support might be unclear, for example when advising on ethical or legal issues. By providing a DMP help desk, responsibilities could be made explicit. Another weakness in DMPs is the lack of assistance in DM cost estimations. Researchers can benefit from semi-automatic assistance and make DM costs explicit and reimbursable from funders.

Funders must monitor DM(P) compliance with their data management policies and can

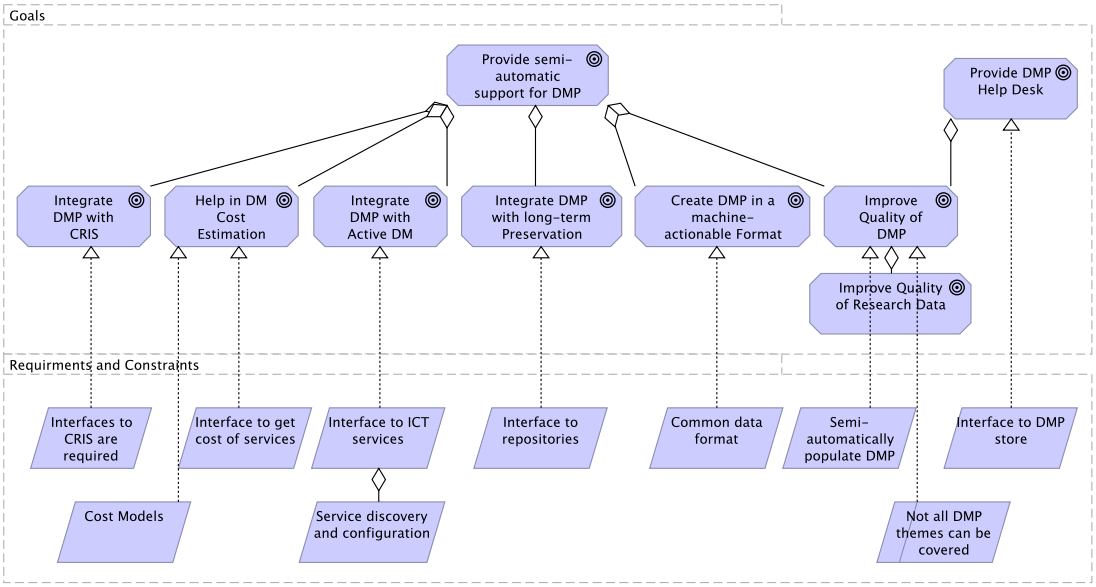


Figure 4.5: Goal realization view showing the goals and derived requirements and constraints.

benefit from an increased efficiency of the review process by implementing semi-automated compliance checks. DMPs in a machine-actionable format can facilitate their automated validation. Support services enable the creation of DMPs with a minimal standard quality by guiding through the DMP process and integrating structured, machine-actionable information.

4.3.2 Goals

We identified the following non-exhaustive list of architectural goals and sub-goals and derived requirements and constraints for achieving them. The goal-realization view illustrated in Figure 4.5 shows the goals and the requirements and constraints derived from them.

- **G1. Provide semi-automatic support for DMP**
  - **G1.1 Integrate DMP with CRIS**
    - R1.1.1 Interface to CRIS is required, e.g. APIs to researcher, project database.
  - **G1.2 Help in DM cost estimation**
    - R1.2.1 Cost Model. Cost models are required that describe the cost of services, e.g. cost of ICT services.

  - **G1.3 Integrate DMP with active DM**
    - R1.3.1 Interface to ICT services. An interface to discover available ICT services relevant for research data management, their QoS, configuration parameters and costs is required.
  - **G1.4 Integrate DMP with long-term preservation**
    - R1.4.1 Interface to repository systems. A standard interface allowing to retrieve metadata from repositories, but also to deposit data in a semi-automated fashion is required.
  - **G1.5 Create DMP in a machine-actionable format**
    - R1.5.1 Common data format. In order to exchange DMP information with external systems, such as funder systems a standard machine-actionable data model needs to be established.
  - **G1.6 Improve quality of DMP**
    - R1.6.1 Semi-automatically fill DMP with structured, machine-actionable information. Semi-automatic support of filling a DMP with meaningful information can ensure the basic DMP quality.
    - C1.6.2 It may not be possible to adequately cover all DMP topics with machine-actionable information. Some questions in a DMP may require user free text responses.

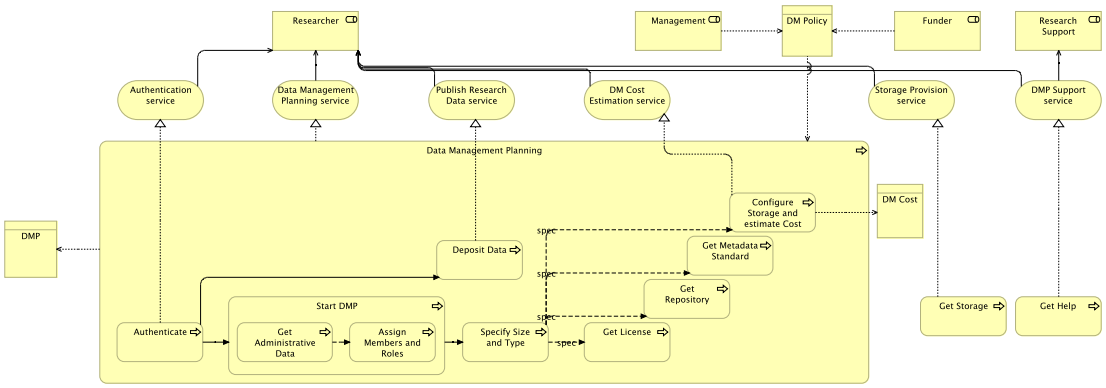
- **G2. Provide DMP help desk**
  - R2.1 Interface to DMP store. In order to provide support to researchers with DMP, the research support staff requires a way to access and view the DMPs.

4.4 Use-Case View

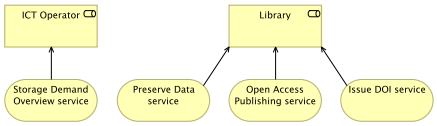
In Chapter 3 we outlined the use cases of a maDMP support system and described the associated workflows as BPMN business processes in more detail (Section 3.2). In this section, we use ArchiMate's business process view to show the relations between these business processes, the services they realize, and the stakeholders they serve in a composite view.

Figure 4.6a and Figure 4.6b show the business process views of our architecture. On the top, the stakeholders and the business services are depicted, for example the *Data Management Planning* service serves the researcher and is realized by the *Data Management Planning* business process. The *Data Management Planning* business process consists of the following sub-processes, which are described in detail in Section 3.2.

1. *Authenticate*



(a) Researcher and research support served by their business proccess services. The *Data Management Planning* process consists of eight sub-processes and outputs a *DMP* business object. Management and funder are involved in the processes indirectly.



(b) ICT operator and library served by their business process services.

Figure 4.6: Archimate's business process views.

2. *Start DMP*
3. *Specify Size and Type*
4. *Configure Storage and Estimate Cost*
5. *Get License*
6. *Get Metadata Standard*
7. *Get Repository*
8. *Deposit Data*

The *Get Storage* business process implements the storage provisioning for the researcher. The *Get Help* business process realizes the *DMP Support service* for the researcher and research support.

The result of the *Data Management Planning* business process is the *DMP* business object, representing the actual data management plan. Data management cost is an output of the *Configure Storage and estimate Cost* sub-process.

Figure 4.6b shows business services of the ICT operator and the library. The *Storage Demand Overview service* serves the ICT operator. The library is served by the following business services.

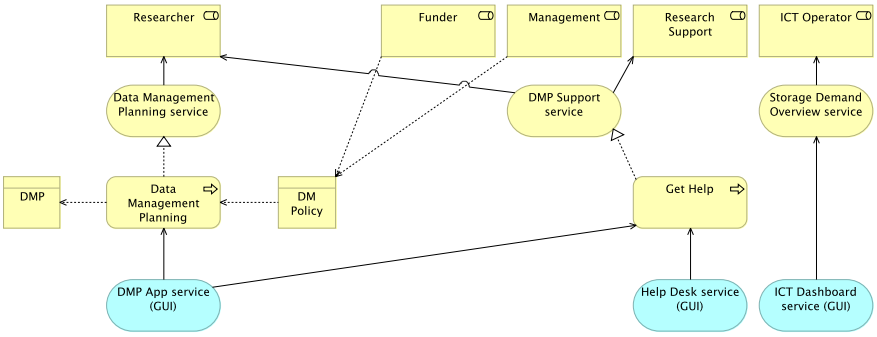


Figure 4.7: Introductory view.

- *Preserve Data service*
- *Open Access Publishing service*
- *Issue DOI service*

4.5 Logical View

Introductory View The introductory view shown in Figure 4.7 gives a very basic overview of the proposed system and shows which Graphical User Interfaces (GUIs) are proposed. Besides the *DMP App* which provides the data management planning GUI for researchers and research support, it shows the *Help Desk* GUI and the *ICT Dashboard* which serves the ICT operator to monitor service demands of computing resources.

Layered View The layered view depicted in Figure 4.8, shows the relations between the elements of the business layer (yellow), the application layer (blue) and the technology layer (green). Each layer provides services which serve the layer above. In this way it can be traced which business process is served by which application service and implemented by which application component. Further, it can be traced which application components are served by which technology service and realized by which technology.

The illustrated application layer only partly shows how application components interact. For example the *CRIS API* serves the *Administrative Data Collector* component. In other words, the *Administrative Collector* component fetches administrative data from the *CRIS API*. However, any details about how application components are integrated and how data flows between application components is missing in this depiction of the architecture but is topic of the implementation view described in Section 4.7.

Some application components in the application layer are grouped in order to show where the proposed components could belong to. We labeled the groups with *CRIS*, *ORCID*, *ICT Operator* and *Library*.

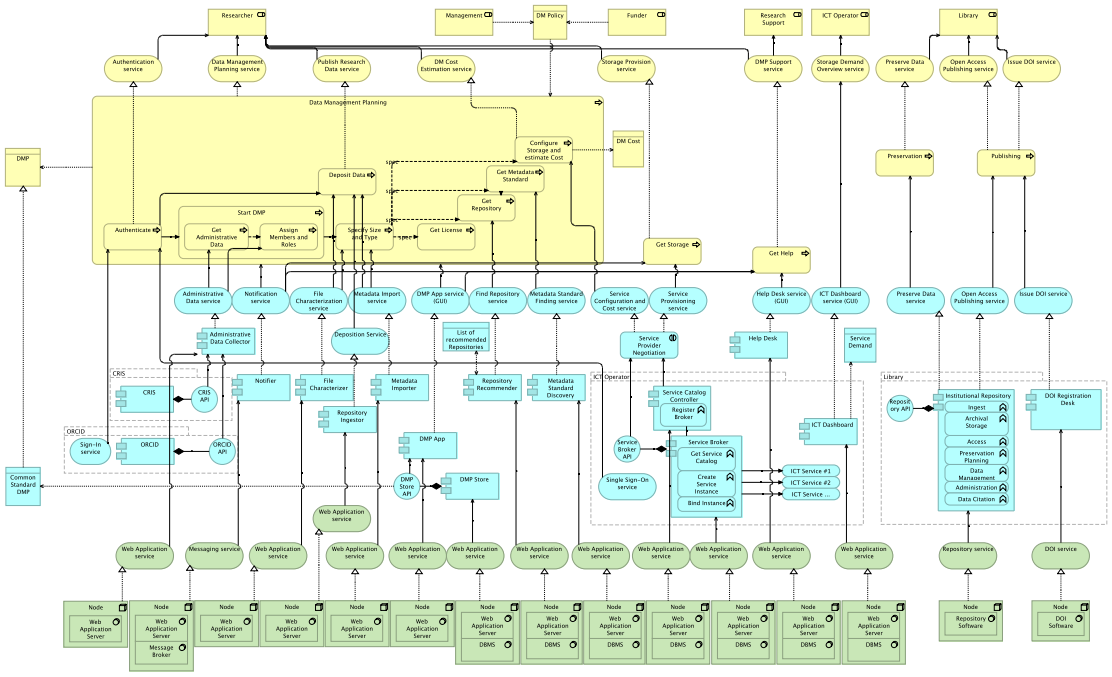


Figure 4.8: Layered view - business (yellow), application (blue), technology (green). See Appendix D on page 117 for the figure in landscape view.

The technology layer (green) at the bottom of Figure 4.8 suggest that application components are realized as web application services. Some web application services are intended to be stateless such as the *Metadata Importer*, while others require a database to persist data such as the *DMP Store*. The relation between technology nodes and application components could change, for example application components could become part of another application component instead of separate web application services. In the current architecture, we propose the web application services listed in Table 4.3.

Further explanation about the application layer is given in Section 4.7.

4.6 Process View

This section describes the runtime behavior of selected workflows and scenarios to stimulate discussions about design questions such as synchronous or asynchronous communication. For this reason, we modeled the UML sequence diagrams of four selected workflows that show interesting aspects. The UML sequence diagrams correspond to the workflows described in Section 3.2.

- Start DMP
- Specify Size and Type

Table 4.3: Proposed web application services for this EA.

Proposed service	Description
DMP Store	Web service maintaining a repository for DMPs providing an API for operations like searching, creating, accessing, modifying, deleting DMPs.
DMP App	Web application serving the DMP GUI and implementing DMP logic.
Repository Recommender	Web service for recommending repositories (API).
Metadata Standard Discovery	Web service for discovering metadata standards (API).
Metadata Importer	Web service for importing metadata (API).
File Characterizer	Web service for uploading and analyzing file samples, identifying their file formats and providing characterization metadata (API).
Notifier	Notification service to deliver messages to users (API).
Administrative Data Collector	Web service for importing administrative data from information systems such as CRIS or ORCID (API).
Repository Ingestor	Web service for ingesting data into a supported repository (API).
Service Broker	Web service for brokering services of an ICT service provider. The broker provides a catalog of services and enables their provisioning (API).
Service Catalog Controller	Web service for registering service brokers from different providers and providing an aggregate catalog of available ICT services (API).
Help Desk	Web application serving the help desk GUI.
ICT Dashboard	Web application serving the ICT dashboard GUI.

- Service Selection and Configuration
- Service Provisioning

4.6.1 Start DMP

The *Start DMP* workflow's sequence diagram depicted in Figure 4.9 shows the inter-process communication between the *DMP App*, *CRIS*, *DMP Store* and *Notifier*.

1. *DMP App* triggers the login process and the user gets directed to the *CRIS* authentication portal.
2. *CRIS* executes the authentication and redirects the user back to *DMP App*.
3. *DMP App* creates or loads a DMP from the *DMP Store*.
4. *DMP App* retrieves administrative data from *CRIS* while the user sets up the administrative information in the DMP.
5. *DMP App* saves the DMP to the *DMP Store*.
6. *DMP App* triggers a notification message to all members using *Notifier*.

4.6.2 Specify Size and Type

The sequence diagram shown in Figure 4.10 depicts the runtime behavior of the *Specify Size and Type* workflow.

1. *DMP App* sends the location (e.g. URL) of a metadata resource to the *Metadata Importer*.
2. *Metadata Importer* harvests the metadata from the location and returns it to *DMP App*.
3. *DMP App* enables the user to upload files to *File Characterizer* for analysis.
4. *File Characterizer* runs a file analysis and returns the result to *DMP App*.
5. *DMP App* saves the result from the file analysis to *DMP Store*.

File transmission, depending on the file size and the connection bandwidth can take a long time. To enable a good user experience the *DMP App* should not block while uploading files, an asynchronous communication is therefore in favor.

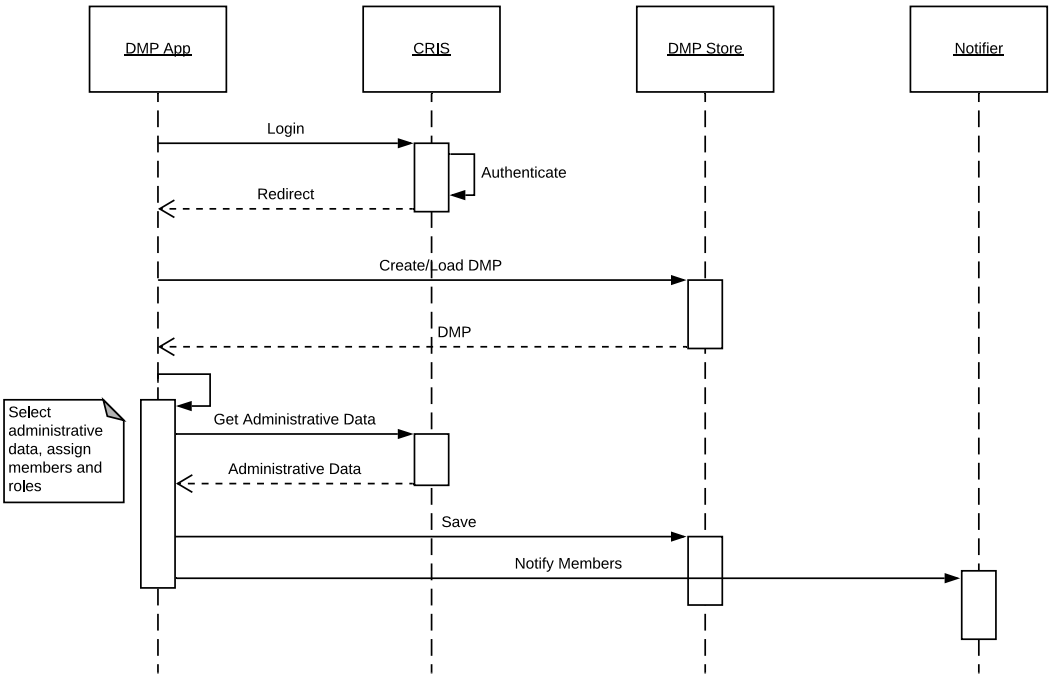


Figure 4.9: Sequence diagram of the *Start DMP* workflow.

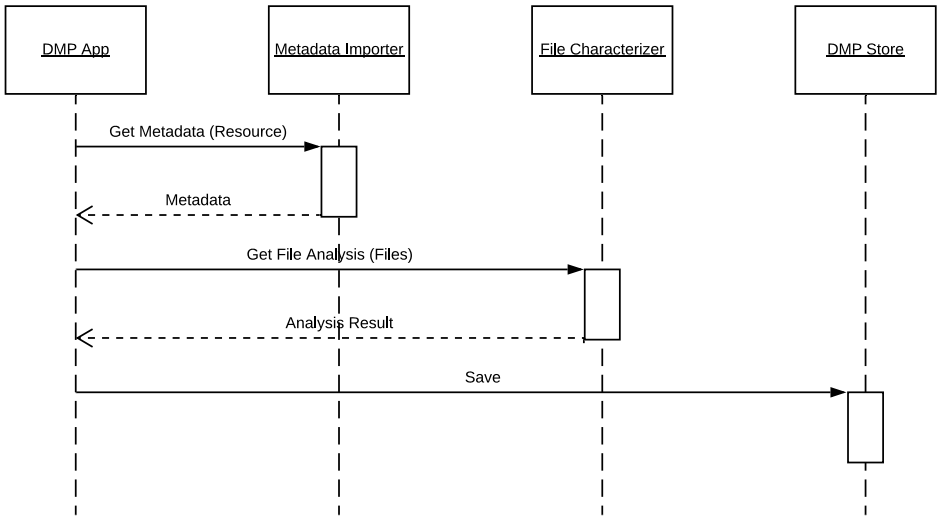


Figure 4.10: Sequence diagram of the *Specify Size and Type* workflow.

4.6.3 Service Selection and Configuration

Figure 4.11 shows the sequence diagram of the *Service Selection and Configuration* workflow. Services such as storage or databases can be configured and provisioned via a standard interface such as the OSBAPI described in Section 2.5.

1. *DMP App* requests a catalog of offered services from the *Service Catalog Controller*.
2. *Service Catalog Controller* requests service catalogs from all registered service brokers.
3. *Service Brokers* return their catalogs with listings of offered services and plans.
4. *Service Catalog Controller* returns an aggregate listing of services and plans from all service brokers to the *DMP App*.
5. *DMP App* presents the user with a catalog of available services and plans which are associated with different QoS and cost.
6. *DMP App* enables the user to select service plans of services and configure them - for example the size of a file-sharing service, or the number of virtual CPUs of a VM.
7. *DMP App* provides the user with a cost estimation of the selected service configurations.
8. *DMP App* saves the service configurations to *DMP Store*.

4.6.4 Service Provisioning

The sequence diagram depicted in Figure 4.12 shows how services are provided to the user via a common interface such as the OSBAPI. The service provider who is responsible for providing the actual service is not depicted in this diagram.

1. Project owner requests the ICT service, e.g. file storage.
2. *DMP App* retrieves the configurations for the previously configured service plans (see Figure 4.11) from *DMP Store*.
3. *DMP App* sends requests for provisioning the services, including configuration, to the respective service brokers.
4. *Service Broker* returns the status of the service provisioning process from the service provider, e.g. service instance was created.
5. *DMP App* requests a service binding from the *Service Broker* when the service instance is ready.

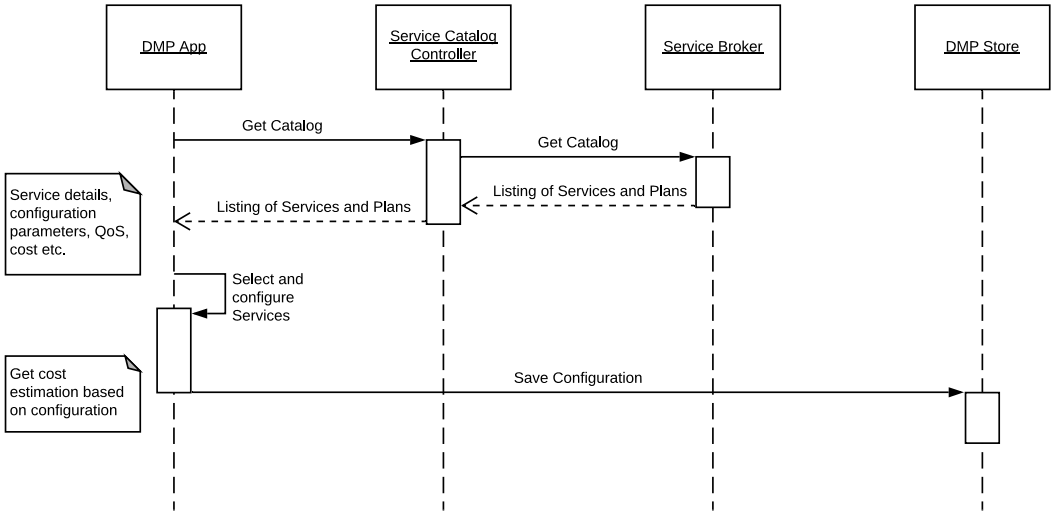


Figure 4.11: Sequence diagram of the *Storage Selection and Configuration* workflow.

6. *Service Broker* returns the binding for the service instance, e.g. containing the access details and credentials for the service instance.

Note that the sequence diagrams presented in this section show the inter-process communication of services at a logical level, but make no assumptions about how services are actually integrated. It is possible to have synchronous calls directly between services or use a message-oriented middleware to exchange messages asynchronously.

4.7 Implementation View

The application structure and co-operation view depicted in Figure 4.13 describes the relationship between application components and shows the flow of information represented by dotted lines. The *DMP App* component, collects information from many other application components. To increase the readability only the incoming information flow of the *DMP App* is shown. The only outgoing information flow from the *DMP App* is directed to the *DMP Store* component to indicate that the information is persisted there.

The *DMP App* component collects information from:

- *Administrative Data Collector*: data from CRIS (project and researcher database) and ORCID
- *Notifier*: incoming notifications
- *File Characterizer*: file analysis results

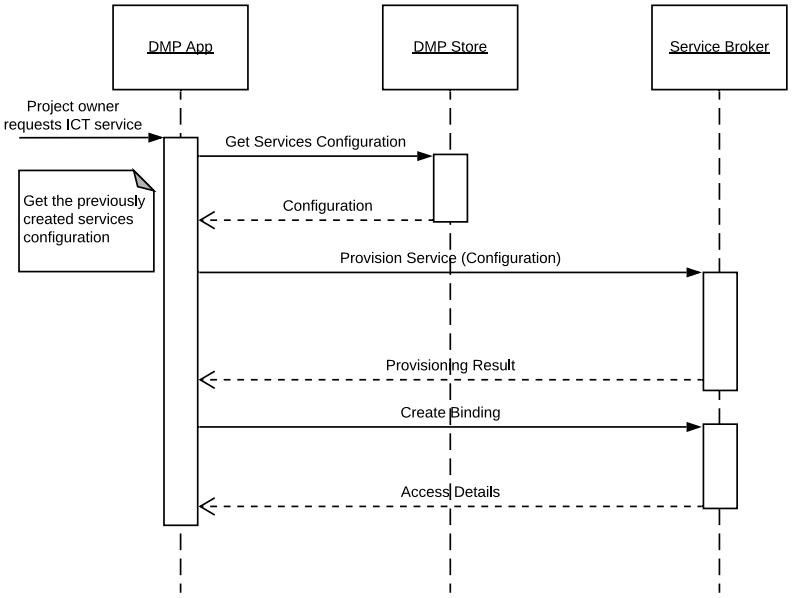


Figure 4.12: Sequence diagram of the *Storage Provisioning* workflow. The Service Broker implements the OSBAPI.

- *Metadata Importer*: metadata harvested from specified resources
- *Repository Recommender*: recommended repositories based on input data
- *Metadata Standard Discovery*: recommended metadata standards based on input data
- *Repository Ingestor*: result of data deposit, e.g. status, DOIs
- *Service Catalog Controller*: information about offered services, see sequence diagram in Figure 4.11
- *Service Broker*: information about service provisioning, see sequence diagram in Figure 4.12

The ICT Operator grouping in the lower right corner of Figure 4.13 shows existing application services provided by the ICT operator such as *Single Sign-on*, *ICT Service #1*, *ICT Service #2* etc. and new application components that are proposed for the system such as *Service Catalog Controller*, *Service Broker*, or *ICT Dashboard*.

- *ICT Dashboard*: visualizes the service demands in the near future, based on the input during data management planning.

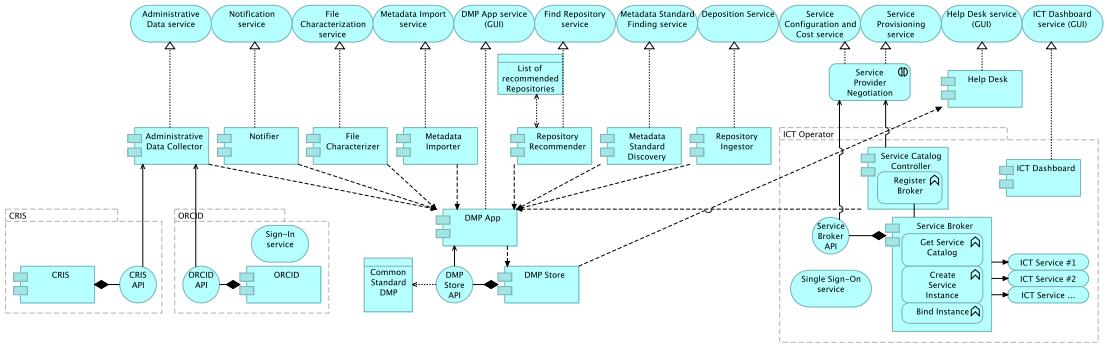


Figure 4.13: Application structure and co-operation view.

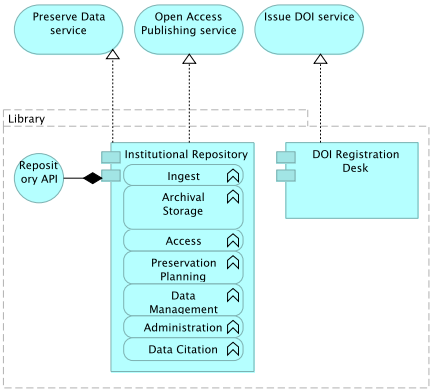


Figure 4.14: Application structure view of the library applications.

- *Service Catalog Controller*: can register service brokers of service providers and acts as an intermediary layer to the service brokers.
- *Service Broker*: implements a standard interface such as the OSBAPI for service discovery and provisioning. It abstracts the specifics of provisioning a service from a service provider in a standard way. The service broker must therefore implement application functions:
  - Get Service Catalog
  - Create/Delete Service Instance
  - Bind/Unbind Service Instance

In the proposed architecture the institutional ICT operator implements a service broker for its existing services, but can also offer further external services by registering service brokers from other service providers available via a common interface such as the OSBAPI.

Figure 4.14 depicts potential application services and components of the library:

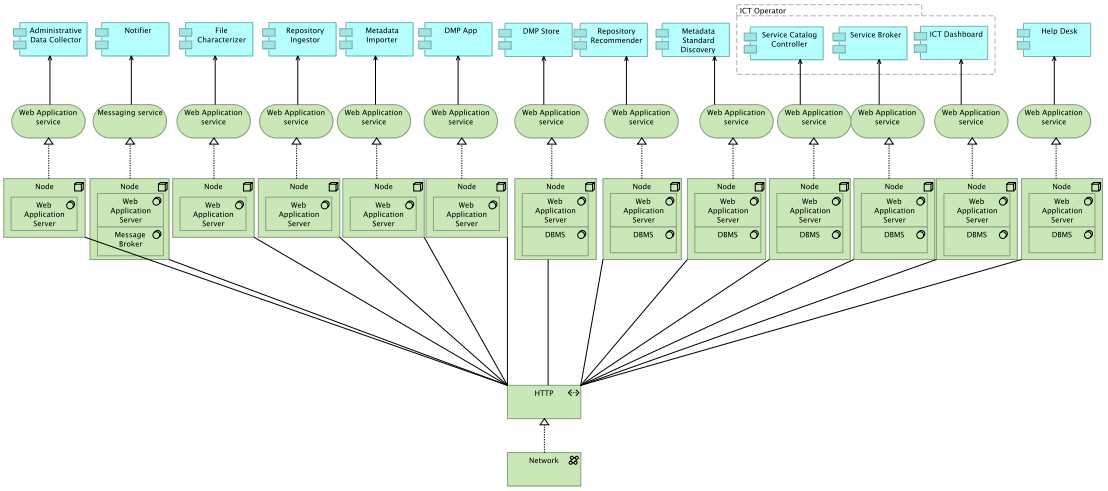


Figure 4.15: Infrastructure usage view depicting a potential service integration over HTTP.

- *Institutional Repository* - repository system, e.g. following the OAIS functional model
- *DOI Registration Desk* - desk issuing DOIs

The proposed architecture is designed to be flexible in the selection of a repository and therefore does not explicitly model the integration of an institutional repository. Also the integration of a potential in-house DOI registration desk is not explicitly modelled.

4.8 Deployment View

The infrastructure usage views depicted in Figure 4.15 and Figure 4.16) show how the application components (blue) are supported by the hardware and software infrastructure (green). These views show that each application component is intended to be realized as a web application service and independently deployable.

The nodes contain system software to enable the web applications to run. To keep the design flexible no specific web application technology is mentioned - the current design allows a polyglot realization of application components which allows to implement web application services in different programming languages and persistence technologies.

The deployment of system software on specific nodes is arbitrary, e.g. a DBMS does not necessarily have to be deployed on the same node as the application server, but could be deployed on a separate node.

The integration of the web application services is indicated in two different ways.

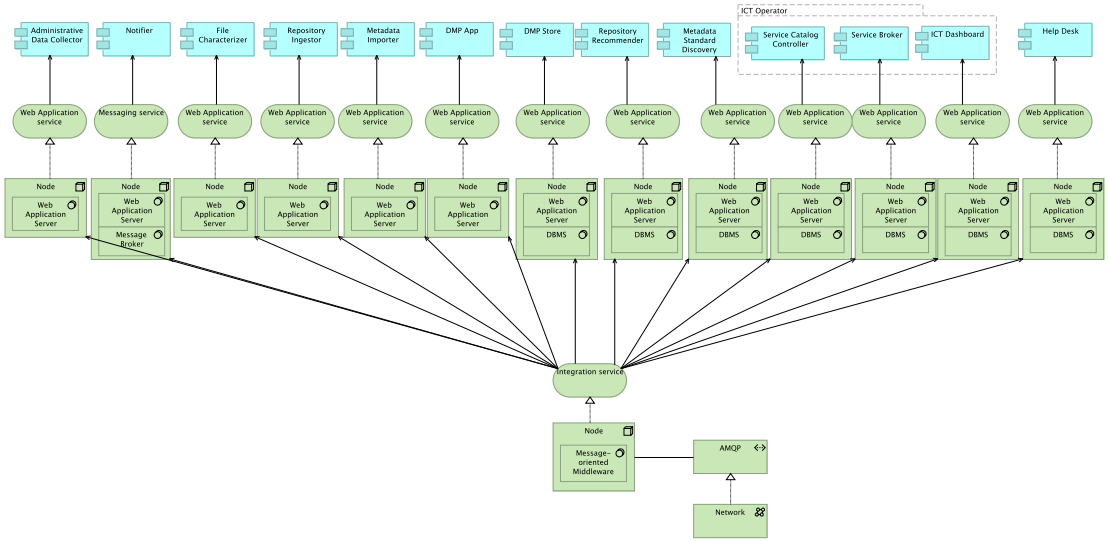


Figure 4.16: Infrastructure usage view depicting a potential service integration via message-oriented middleware.

- Figure 4.15 indicates an integration of web application services via HTTP. In this scenario web application services have a direct coupling and use synchronous calls, e.g. via HTTP REST interface to communicate with each other.

- Figure 4.16 indicates an asynchronous integration of web application services using a message-oriented middleware. Web application services can be loosely-coupled and communicate in an event-driven manner by publishing messages in a message queue, for example via AMQP, which other services can subscribe to.

Figure 4.15 and Figure 4.16 show two types of system integration, whereby a combination of both types of integration is also possible.

4.9 Summary and Discussion

In this chapter we used EA modelling to describe the proposed architecture for an institutional maDMP support system by using multiple, concurrent views. We defined architectural goals and associated requirements and constraints by assessing the drivers for change of an institutional RDM infrastructure. We integrated the workflows derived from community use cases into the business process layer of the architecture. For the business processes and sub-processes, we derived application services and components that can implement them. By arranging architectural components around business capabilities, we achieved a modular architecture that is polyglot and enables self-contained components to be independent of programming language and technology.

We identified thirteen application services that are required to implement the business processes. The services vary in complexity and form. Some services are intended to provide a GUI, while others offer an API. The system is designed to be deployed on an institutional level, but some of the services could be shared with other institutions or the public. For example, the *Repository Recommender* service that consumes DMP and policy information in a standard format can be deployed publicly. The same applies for the *Metadata Standard Discovery*, the *File Characterizer*, the *Metadata Importer*, or the *Repository Ingestor* services under certain conditions. The *Help Desk* service could be shared between institutions.

In order to integrate ICT services relevant to RDM into data management planning, we propose a *Service Broker* that implements the OSBAPI, which can abstract any service and offer a standard interface for service discovery, configuration and provisioning.

The EA describes the relationships between components of the business, the application and the technology layers of the architecture, ensuring the transparency and traceability of design decisions.

CHAPTER 5

Implementation

In this chapter we describe an implementation of the proposed machine-actionable DMP support system in an institutional context. We choose the TU Wien as an example institution to show how the proposed system can be embedded in the system and service landscape of an institution. The implementation helps to evaluate the efficiency and effectiveness of such a system.

This chapter is structured as follows. First, we introduce the TU Wien case study and describe the systems, services and stakeholders present at the institution. We then present the DMap tool, which implements selected automated workflows using the TU Wien infrastructure and other systems and services. Subsequently, we describe the service broker implementation for ICT services available at the TU Wien and conclude with a summary and discussion.

5.1 TU Wien Case Study

In this section we describe the context of our implementation. We use the TU Wien as a case study for our implementation. TU Wien is a major university in Austria with more than 27,000 students, around 5,000 employees and eight faculties. It has departments, information systems and ICT services that are relevant to our machine-actionable DMP use cases.

The automated workflows described in Section 3.2 involve three main stakeholders, namely the researcher, research support and ICT operator. At the TU Wien, DMP-related research support is offered by the Center for Research Data Management and the European and International Research Support (EIRS) team. TU.it is the local ICT operator that provides ICT services for employees and researchers. The services offered range from storage solutions, virtual server hosting to the provision of HPC infrastructure and more.

Table 5.1: TU.it data storage service solutions.

	TUfiles	Cloud storage	
		TUownCloud	TUproCloud
Technology	Windows Server	ownCloud	
Clients	SMB clients	ownCloud clients for all platforms	
Availability	98.5% and 97% in non office hours		
Backup	Four daily snapshots, one daily on weekends, 64 versions at maximum	Daily disaster recovery backup	
Sharing with externals	no	no	yes
Security	Self-encrypting volumes, access only via VPN	Self-encrypting volumes	
Storage space	Up to some TB	20GB	Up to some 100GB
Costs	0.03€/GB/quarter	free	0.03€/GB/quarter

5.1.1 TU.it Services

TU.it operates an in-house data center and provides several services for research data management. In the following we describe the TU.it services for the storage and sharing of research data and the hosting of virtual servers. The list of described services is non-exhaustive, and TU.it continuously expands its service portfolio. The service descriptions are based on the information provided on the TU.it website[1].

Data Storage

TU.it offers three data storage service solutions, namely TUfiles, TUownCloud and TUproCloud. TUfiles is a storage service to store data on a redundant, central and highly available network drive. For file synchronization and sharing TU.it offers the services TUownCloud and TUproCloud, which both are instances of the open source software ownCloud[2]. While TUownCloud comes with 20GB free of charge for all employees, TUproCloud costs 0.03€ per GB and quarter and adds the possibility to share data with TU external collaborators. Table 5.1 gives an overview of data storage solutions, their characteristics and costs, which are provided by TU.it.

Virtual Server Hosting

TUhost is a hosting service for virtual servers with which Virtual Machines (VMs) can be operated on TU.it's central and highly available virtualization platform. The service is based on the VMware ESXi hypervisor and the SANsymphony storage virtualization.

[1] https://www.it.tuwien.ac.at/
[2] https://owncloud.org/

Table 5.2: TU.it virtual server hosting solution.

	TUhost
Technology	VMware ESXi, SANsymphony
VM Operating system	CentOS/Debian/Windows Server
Characteristics	High availability, moderate resources
Backup	Full disaster recovery, daily backups, 30 days retention time
Security	User responsibility
Costs	
vCPUs	8.00 €/CPU/quarter
RAM	8.00 €/GB/quarter
DiskSys	0.10 €/GB/quarter
DiskData	0.10 €/GB/quarter
DiskHighPerf	0.25 €/GB/quarter

For example, the service can be used to host application servers, code repositories, or databases. The VMs can be ordered with configurable settings for the operating system, computing power and storage size. Table 5.2 gives an overview on the TUhost service characteristics, the configuration parameters and the associated costs.

Service Discovery and Provisioning

TU.it services can be discovered on their website. The process of ordering most of the TU.it services involves filling out an order form on their website, where access, billing and service configuration are specified. Then, an authorized member of the selected billing group gets notified and can decide if the service request will be approved. If approved, a ticket is created in the TU.it ticket system and an administrator can process the service request and contact the customer to discuss the order. After discussing order details, the administrator creates the service instance and notifies the customer when the service is ready.

5.1.2 TU Wien Information Systems and Services

The TU Wien Information Systems and Services (TISS)[3] is the central information system to ensure university operations. TISS includes a searchable address book of employees, students and organizational units. TISS also contains a project database to manage research projects and their funding. Both of these two sub-systems contain relevant information for DMPs. The address book serves as a source for personal information about researchers and other employees, such as person ID, email or affiliation. The project database contains relevant project and funding information, such as project ID, title,

[3] https://www.tiss.tuwien.ac.at/

duration, funder or grant ID. TISS provides a public Representational State Transfer (REST) API to fetch information from the address book and the project database.

5.1.3 Assumptions

Before we go into detail about the implementation of our TU Wien case study, we would like to describe the underlying assumptions. The implementation is not a general tool for machine-actionable data management planning, but is a proof-of-concept tool tailored for the TU Wien case study. Therefore, we made the following assumptions.

- The person creating the DMP is a member of the TU Wien.
- TU Wien members only will contribute to the data management.
- The projects for which DMPs are created are registered in the TISS project database, see Section 5.1.2.
- The researchers use ICT services provided by TU.it, see Section 5.1.1.
- Repositories where researchers plan to deposit data are indexed in the re3data[4], see Section 5.2.6.

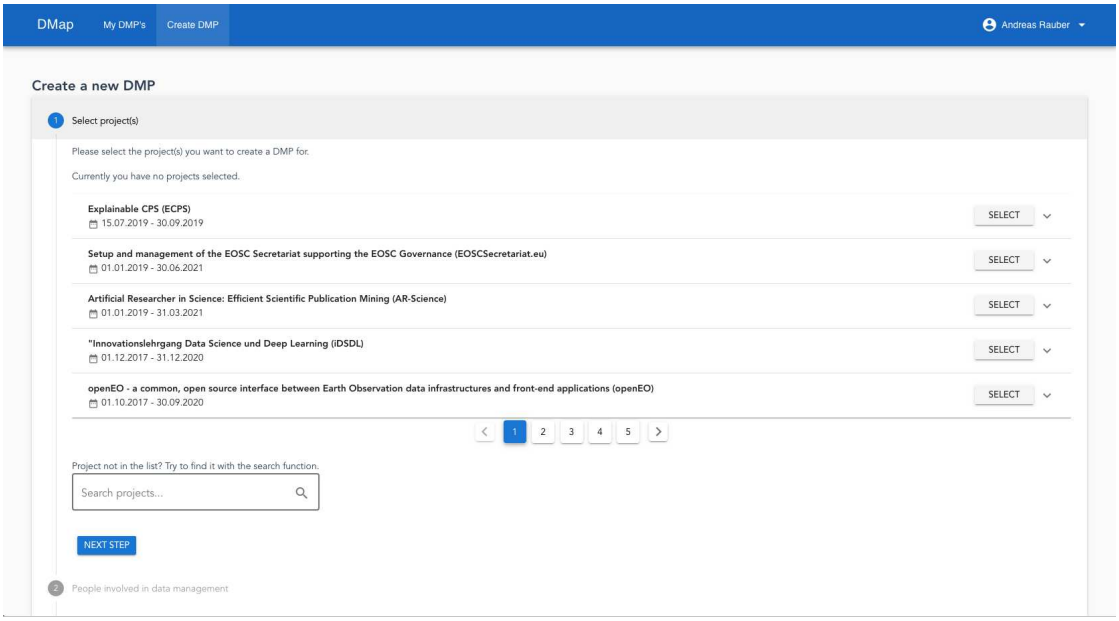
5.2 DMap Tool

In order to show the applicability of the Enterprise Architecture described in Chapter 4 in a real scenario, we developed a proof-of-concept tool named DMap. DMap uses the infrastructure of the TU Wien and its systems and services, as described in the introduction to the case study in Section 5.1. In DMap we implemented selected use cases from the set of workflows described in Section 3.2. DMap is the implementation of the *DMP App* application service and related application components described in the EA and therefore realizes the service used by researchers and research support staff. Other application services described in the EA such as the *ICT Dashboard* or the *Help Desk* are not implemented as part of this work. The DMap user interface was implemented according to the mockup design described in Section 3.3. DMap is open source software, licensed under the MIT license and available on GitHub[5]. Figure 5.1 shows screenshots of DMap in action. Further DMap screenshots can be viewed in the slides [Obl19] resulting from the presentation of DMap to the Open Science community at the 14th Research Data Alliance plenary in Helsinki, October 2019.

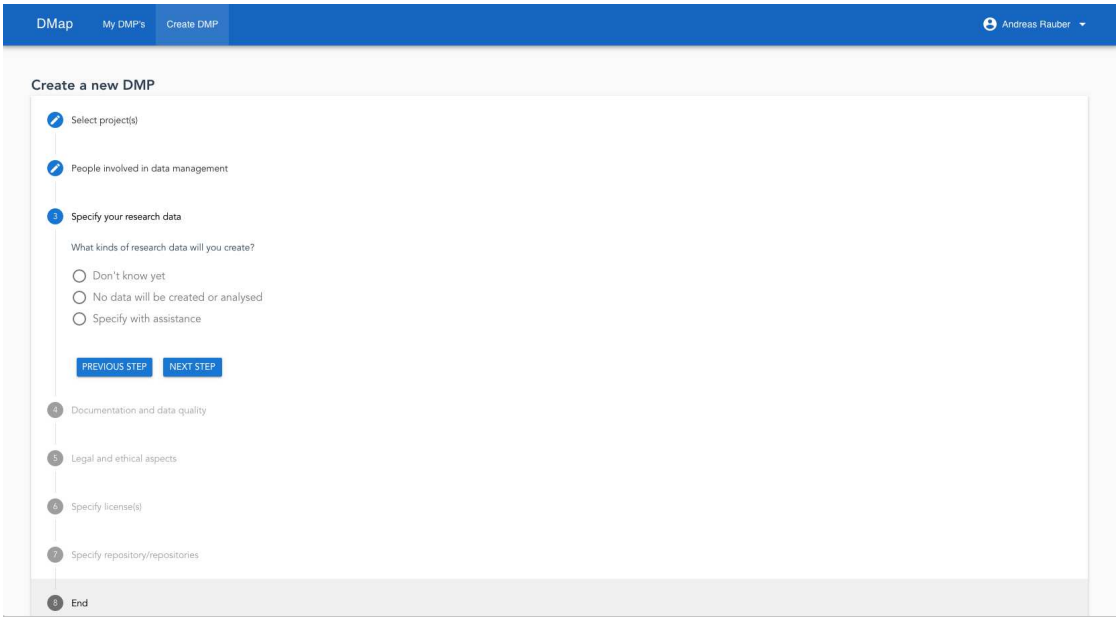
5.2.1 Technical Overview

DMap is the implementation of the *DMP App* application service and related application components described in the EA in Chapter 4. As indicated in the EA, DMap is separated

[4]https://www.re3data.org/
[5]https://github.com/oblassers/dmap



(a) Project selection in DMap.



(b) Steps to create a DMP in DMap.

Figure 5.1: The machine-actionable Data Management planning application (DMap) is a proof-of-concept tool to demonstrate selected features of the proposed architecture.

into two decoupled services - a backend and a frontend service. The communication between the frontend and backend is realized as a resource-based, synchronous communication over Hypertext Transfer Protocol (HTTP). Hence, the backend provides a REST API which serves as a gateway to various kinds of resources, such as DMPs stored in the database or project information located at the external TISS service. JSON is used as the payload format for the transfer of structured information between the backend and the frontend. The frontend runs in a web browser and provides the UI to the user.

The backend is implemented as a Spring Boot[6] application with embedded Apache Tomcat servlet engine using Java 8. The communication with the DMP database is done via an abstraction data access layer provided by Spring Data. Currently, DMap uses the document database MongoDB[7] to persist DMPs, but it can easily be replaced with any other datastore due to the used data access abstraction layer.

The fronted has been developed with the JavaScript framework Vue.js[8] and makes use of several libraries to provide a scalable and maintainable architecture. For example, Vue Router is used for client-side routing between different views, which enables us to create a modern single page web application. The UI is composed of self-contained and reusable Vue components that contain Hypertext Markup Language (HTML) template code, JavaScript code and Cascading Style Sheets (CSS) style information. A central Vuex state store is used to maintain a consistent state of the frontend application. To communicate with the backend a HTTP service using the Axios library has been implemented.

An overview of the DMap backend and frontend architecture is depicted in Figure 5.2.

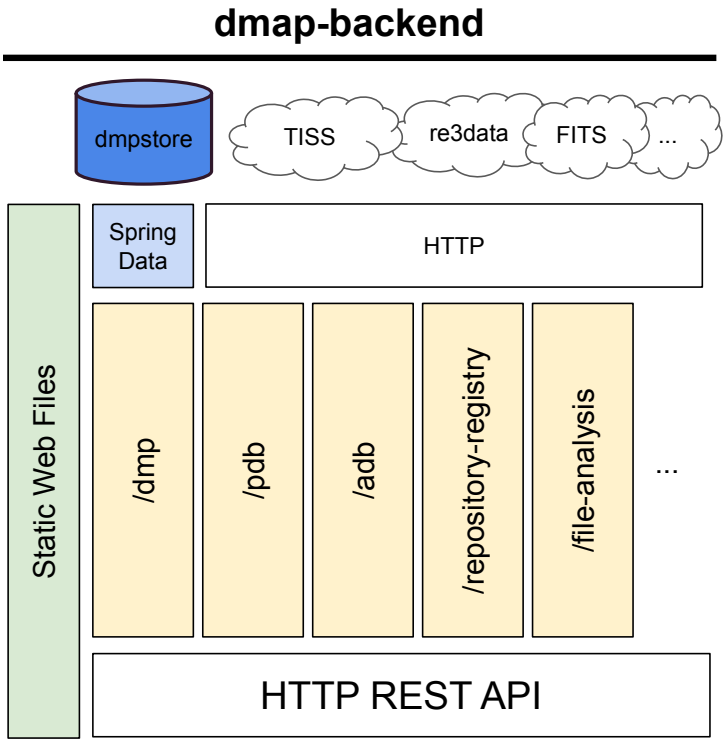
5.2.2 Implemented Workflows Overview

The Enterprise Architecture described in Chapter 4 outlines how a maDMP support system could be implemented. The foundation of the architecture are the business processes described in Section 3.2. However, within the scope of this thesis we did not implement all parts of the architecture that realize these business processes. Therefore, we chose a subset of the business processes and describe their implementation in DMap. Table 5.3 provides an overview of which business processes have been implemented. In the following sections we describe the implementation of the selected workflows in DMap.

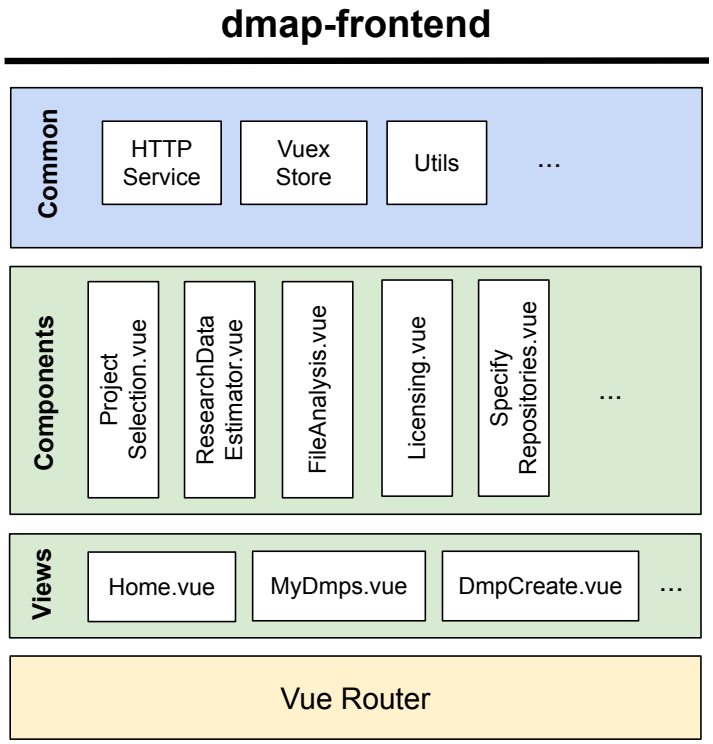
5.2.3 Start DMP

The *Start DMP* workflow was described in Section 3.2.2 on page 27. In DMap we implemented the selection of one or more research projects the DMP is being created for and the selection of people involved in data management as well as their assignment of contribution roles. For this purpose we created endpoints in the DMap backend which serve as gateways to query resources from the public TISS HTTP REST API.

[6] https://spring.io/projects/spring-boot
[7] https://www.mongodb.com
[8] https://vuejs.org



(a) DMap backend.



(b) DMap frontend.

Figure 5.2: Overview of the DMap backend and frontend architecture.

Table 5.3: Workflows implemented in DMap. The workflows are described in Section 3.2 of the Requirements Engineering Chapter 3. The implementation of the selected workflows is described in this chapter.

Workflow	Implemented in DMap
Start DMP	yes
Specify Size and Type	yes
Get Cost and Storage	see TU.it Service Broker, Section 5.3
Get License	yes
Get Metadata Standard	no
Get Repository	yes
Deposit Data	no
Get Help	no

Based on the user identifier (OID) and the users' memberships in organizational units of the TU Wien, we implemented a simple algorithm to suggest research projects for which the user might want to create a DMP. We query for research projects where the user is project leader and combine the result set with projects of the institutions where the user is a member of. Before returning the list of research projects it is sorted in a way that first lists the projects where the user is project leader and then the projects with their begin dates in descending order, listing the latest projects first.

When the user selects a project, further project details are retrieved from the project database via the TISS API. The project details contain information about the project members, the research areas or the project funding. The user automatically gets presented with a list of project members fetched from the TISS address book and can assign contribution roles such as Contact Person, Data Manager or Data Curator. As a vocabulary for the contribution roles we use the DataCite Metadata schema specification [Dat19] for contributorType.

5.2.4 Specify Size and Type

The *Specify Size and Type* business process was described in Section 3.2.3 on page 28. In DMap we implemented two methods to estimate and specify the data that will be created during the research project - a manual and an automatic approach. The methods can be combined to obtain a compound research data estimation.

Manual Research Data Specification

When estimating research data manually, the user can add new estimations and specify the type of data by selecting from a list of types. We use the controlled vocabulary for the contentType from the re3data metadata schema [RVU+15] because it presents a comprehensive, yet manageable list of distinct data type categories. The data type

Table 5.4: Controlled vocabulary to specify the type of research data in DMap [RVU+15].

Type of Data	Examples of File Formats
Standard office documents	text documents, spreadsheets, presentations
Networkbased data	websites, email, chat history, etc.
Databases	DBASE, MS Access, Oracle, MySQL, etc.
Images	JPEG, JPEG2000, GIF, TIF, PNG, SVG, etc.
Structured graphics	CAD, CAM, 3D, VRML, etc.
Audiovisual data	WAVE, MP3, MP4, Flash, etc.
Scientific and statistical data formats	SPSS, FITS, GIS, etc.
Raw data	device specific output
Plain text	TXT in various encodings
Structured text	XML, SGML, etc.
Archived data	ZIP, RAR, JAR, etc.
Software applications	modelling tools, editors, IDE, compilers, etc.
Source code	scripting, Java, C, C++, Fortran, etc.
Configuration data	parameter settings, logs, library files
Other	-

categories used in DMap are shown in Table 5.4. To estimate the storage size of the research data, the user can choose from a list of size ranges as displayed in Table 5.5.

Automatic Research Data Specification

With the automatic approach to specifying research data in DMap, the user can upload files for analysis to determine file formats and sizes. For this purpose, we use the File Information Tool Set (FITS)[9], a file analysis tool from the field of digital libraries. We run FITS as a web service in a Docker container and forward the files for analysis from DMap. Depending on the configuration, FITS internally runs several file analysis tools such as JSTOR/Harvard Object Validation Environment (JHOVE)[10], Digital Record and Object Identification (DROID)[11] or Apache Tika[12] and consolidates the various analysis results into one standard Extensible Markup Language (XML) summary which is further processed by DMap. Figure 5.3 depicts the file analysis process in FITS.

FITS can not only identify file formats, but also extract technical metadata and validate file formats. However, in our use case we are interested in file format identification and file size only. Most of the tools encapsulated in FITS provide information on the file Multipurpose Internet Mail Extensions (MIME)-type. Additionally, DROID provides more precise information on the identified file format by returning the PRONOM

[9] https://projects.iq.harvard.edu/fits
[10] http://jhove.openpreservation.org/
[11] http://digital-preservation.github.io/droid/
[12] http://tika.apache.org/

Table 5.5: Controlled vocabulary to estimate the size of research data in DMap.

Size Ranges for Research Data
< 100 MB
100 - 1000 MB
1 - 5 GB
5 - 20 GB
20 - 50 GB
50 - 100 GB
100 - 500 GB
500 - 1000 GB
1 - 5 TB
5 - 10 TB
10 - 100 TB
100 - 500 TB
500 - 1000 TB
> 1 PB
Don't know

Persistent Unique Identifier (PUID)[13]. The PUID uniquely identifies a file format within the PRONOM registry which indexes more than 1600 different file formats. In DMap we extract the desired information from the FITS response and send it to the frontend to show the analysis result to the user. Listing 5.1 shows the extracted file analysis result for an uploaded image file containing all the required information such as file format name, MIME-type, PUID and file size in bytes.

Listing 5.1: File analysis result for a uploaded image file containing format name, MIME-type, PUID and file size in bytes.

```
{
    "format": "Portable Network Graphics",
    "mimeType": "image/png",
    "formatIdentifier": {
        "id": "fmt/13",
        "type": "puid"
    },
    "size": 191504
}
```

Group Research Data Specifications into Datasets

Manually or automatically created research data specifications can be grouped into datasets in DMap. A dataset represents a logical collection of research data and can

[13] https://www.nationalarchives.gov.uk/aboutapps/pronom/puid.htm
[14] https://projects.iq.harvard.edu/fits/fits-processing

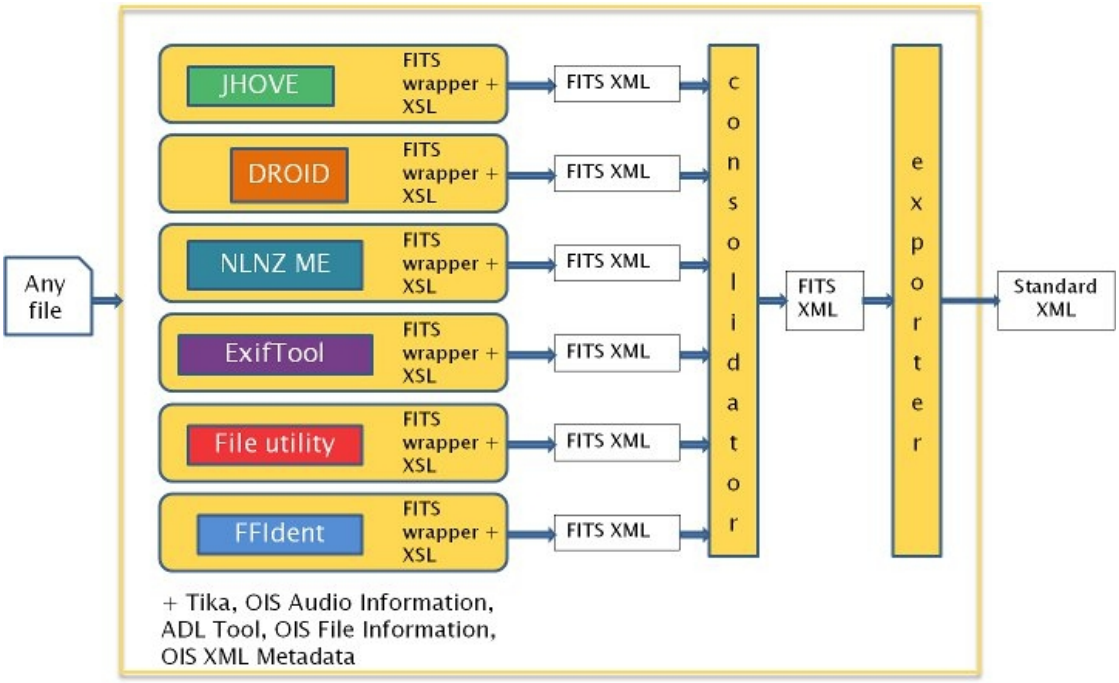


Figure 5.3: Overview of FITS processing[14]. Depending on the configuration multiple tools such as JHOVE, DROID or Apache Tika analyze the files. The various file analysis results are consolidated into one standard XML summary.

contain different types of research data. For example, a dataset representing software might contain source code, configuration data and a database. In DMap the user can create custom names for datasets and assign the specified research data to the datasets. The total storage sizes of the datasets are automatically calculated and presented to the user.

The logical grouping of research data specifications into datasets plays a role in the following steps of creating a DMP in DMap. For example, planned licenses can be assigned to datasets or repositories are selected for individual datasets. Datasets can also be flagged if they contain personal or sensitive data.

5.2.5 Get License

The *Get License* workflow was described in Section 3.2.5 on page 32. In DMap, we integrated the European Data Infrastructure (EUDAT) license selector tool[15] into the frontend. The JavaScript-based EUDAT license selector tool provides a wizard-like interface which guides the user through a set of questions to narrow down the applicable licenses for the publication of research data. The flow chart of the decision questions is depicted in Figure 5.4. The EUDAT license selector tool supports 22 common licenses

[15]https://eudat.eu/services/userdoc/license-selector

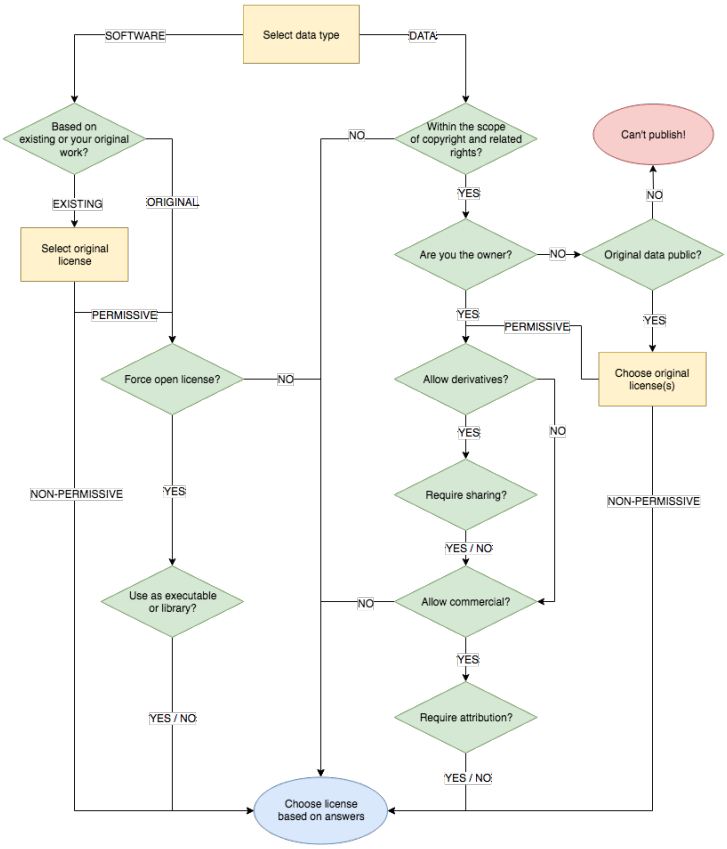


Figure 5.4: Decision flow diagram of EUDAT license selector tool[16].

for the publication of software or data. If none of these are applicable for the data at hand, the user can set their own license in DMap.

In DMap, the user can mark which datasets should be kept closed and which should be published. For the datasets to be published, the user can select a planned license and publication date. The user can either select a license directly from a list or use the EUDAT license selector tool for guidance if needed.

5.2.6 Get Repository

The *Get Repository* workflow was described in Section 3.2.7 on page 33. In the DMap implementation, the user can search for suitable repositories and specify which datasets should be deposited in which repositories. For this purpose, we implemented an integration of the re3data[17] in DMap.

[16] https://eudat.eu/services/userdoc/license-selector
[17] https://www.re3data.org/

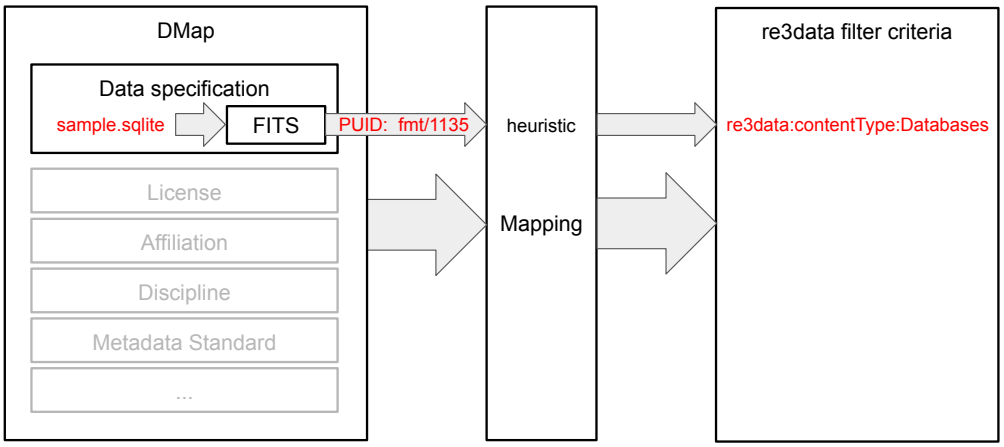


Figure 5.5: Mapping of structured information captured in DMap to re3data filter criteria.

Re3data is a service of DataCite and provides a curated, searchable and filterable online registry that indexes more than 2400 research data repositories. Repository owners can register their repository with re3data and provide extensive metadata [RVU+15], such as the content types the repository supports, the type of certification it has, what PID system it uses, which metadata standards are supported or which data access conditions apply. This metadata associated with each entry in re3data enables the search for repositories that meet certain criteria. Re3data provides a public HTTP REST API[18] to query the registry for repositories matching filter criteria and do a full-text search.

The DMap frontend enables the user to apply the filters supported by the re3data API and to enter text for a full-text search in the registry. After the DMap backend queries the re3data API, the search results are then presented to the user in the frontend. The user can view the repository details to get more information about the matching repositories and select repositories she or he plans to use for storing the research data. If datasets were specified as described in Section 5.2.4 the user can further specify which datasets are planned to be stored in the selected repositories and how long they should be available.

The structured information captured in DMap such as the specified datasets or licensing information present an opportunity to reuse this information to automatically apply filters and query the repository registry as described in Section 3.2.7. This requires a mapping from the structured information captured in DMap to the registry filters. For example, a *.sqlite file can be uploaded to DMap. FITS automatically identifies the file format as described in Section 5.2.4, which is mapped to the suitable content type filter in re3data as illustrated in Figure 5.5. To create a mapping from the more than 1600 file formats identifiable by their PUID to the set of re3data content types heuristics may be applied. However, the mapping must be manually established first and has not been implemented in DMap as part of this work.

[18] https://www.re3data.org/api/doc

5.2.7 Export to RDA DMP Common Standard

DMap can export the DMP in the machine-actionable format specified by the RDA DMP Common Standard, as described in Section 2.3.3. The exported maDMP enables the structured and standardized exchange of DMP-related information and thus facilitates interoperability between research systems. In this section we describe how the structured information captured in DMap is mapped to the data fields of the RDA DMP Common Standard for the implementation of the export function. For this we use the latest version of the RDA DMP Common Standard[19], v1.0.

DMap implements selected workflows and hence does not capture all information described by the RDA standard. We therefore describe which parts of the RDA standard are covered by the implementation in DMap and which are missing.

Mapping

The RDA DMP Common Standard specifies the data type for each of the data fields. For some of the data fields the data type can be primitive such as a string or a numeric value, whereas values for other fields are restricted by controlled vocabularies. For example, the allowed values for the field `dmp.ethical_issues_exist` are restricted to the values `{yes, no, unknown}`. Wherever possible in DMap we use the controlled vocabularies defined by the RDA standard. However, depending on the origin of the information captured, it is necessary to apply a mapping to be compliant with the standard. In the following we describe the peculiarities of the RDA standard export in DMap.

In DMap we integrated external systems and services such as TISS, FITS, EUDAT License Selector or re3data. The structured information sourced from these systems and services follow their own schemes specified by their APIs. In most of the cases the collected information aligns very well with the RDA standard. Table 5.6 and Table 5.7 show the data fields of the RDA DMP Common Standard v1.0 and the corresponding origins of the information captured in DMap and used for the standard export. An example of a machine-actionable DMP that follows the RDA standard v1.0 and was exported from DMap is listed in the Appendix A.

The RDA standard supports multiple types of identifiers for entities such as for the DMP itself, contributors, funders, grants or datasets. For example, the identifier for a contributor can be of type `{orcid, isni, openid, other}`. If common identifiers are not available, custom identifiers can be used - the RDA standard supports this with an identifier type `other`. We use the identifier type `other` for several entities in DMap. For example, we use TISS identifiers for contributors and research projects which link to the respective resources within TISS. Since DMap does not integrate the ORCID API and only some of the person profiles in TISS contain an ORCID, we use the TISS ID. This similarly applies for the funder and grant identifier - DMap does not integrate the

[19] https://github.com/RDA-DMP-Common/RDA-DMP-Common-Standard/tree/v1.0

Table 5.6: Data fields (1/2) and their origins of information to export a maDMP in the RDA DMP Common Standard v1.0 [MWN19] from DMap.

RDA DMP Common Standard v1.0 fields				Origin
dmp_id				DMap
title				DMap
description				DMap
language				DMap
created				DMap
modified				DMap
ethical_issues_exist				DMap
ethical_issues_report				DMap
ethical_issues_description				DMap
contact	contact_id	identifier		TISS API
		type		
	mbox			TISS API
	name			TISS API
contributor	contributor_id	identifier		TISS API
		type		
	name			TISS API
	mbox			TISS API
	role			DMap
project	title			TISS API
	description			TISS API
	start			TISS API
	end			TISS API
	funding	funder_id	identifier	TISS API
			type	
		funding_status		missing
		grant_id	identifier	TISS API
			type	
cost	title			missing
	description			missing
	value			missing
	currency_code			missing

Table 5.7: Data fields (2/2) and their origins of information to export a maDMP in the RDA DMP Common Standard v1.0 [MWN19] from DMap.

RDA DMP Common Standard v1.0 fields				Origin
dataset	dataset_id			DMap
	title			DMap
	type			DMap
	description			DMap
	issued			missing
	keyword			missing
	language			missing
	personal_data			DMap
	sensitive_data			DMap
	preservation_statement			missing
	data_quality_assurance			missing
	metadata	metadata_standard_id	identifier	missing
			type	missing
		description		missing
		language		missing
	security_and_privacy	title		missing
		description		missing
	technical_resource	name		missing
		description		missing
	distribution	title		DMap
		description		missing
		format		FITS
		byte_size		FITS
		data_access		DMap
		access_url		missing
		download_url		missing
		available_until		DMap
		license	license_ref	EUDAT Selector
			start_date	DMap
		host	title	re3data API
			url	re3data API
			description	re3data API
			storage_type	DMap
			geo_location	missing
			certified_with	re3data API
			pid_system	re3data API
			availability	missing
			backup_frequency	missing
			backup_type	missing
			support_versioning	re3data API

Crossref funder registry[20] but uses the funder name and grant identifier stored in the TISS project database.

For some data fields of the RDA standard, constraints are relaxed and certain types are recommended but not enforced. In the RDA standard the value for the `dmp.dataset.type` is open and therefore we can use the set of research data types from the data specification step in DMap, see Section 5.2.4. The field `dmp.dataset.distribution.format` is also open and we follow the standard recommendation by using the MIME-type. However, we could also use the PUID captured by the FITS file analysis if a more precise format identification is required.

The RDA standard specifies controlled vocabulary for some data fields that do not exactly match the information captured in DMap. In the `dmp.dataset.distribution.host` entity the RDA standard prescribes the allowed values for the `certified_with`, `pid_system` and `support_versioning` fields. When selecting a repository in DMap, the repository attributes are fetched from the re3data API which uses another controlled vocabulary[21] for the same fields. However, most data field values are very similar, differ in spelling or contain a different range of values.

To export to the RDA standard, some of the information captured in DMap must be reduced. For example, when specifying the research data in DMap, the user can estimate the size of datasets by using size ranges. The RDA standard uses a single value to describe the size of the datasets in bytes. Hence, we map the upper bound of the DMap size estimation to the size value of the RDA standard.

Not all workflows have been implemented in DMap and therefore the information collected is not complete in order to be able to fill in all fields of the RDA standard. For example, the cost estimation and metadata standard selection have not been implemented and hence are missing in the exported maDMP. The service broker (Section 5.3) can provide information on the cost of ICT services used for data management. In this work, however, we have not integrated the TU.it service broker into DMap and therefore marked this information as missing. Also, we do not capture the free-form texts that describe technical resources and security and privacy matters of the RDA standard. The missing data fields are marked in red in Table 5.6 and Table 5.7. In some cases the reason for a missing data field is due to the lacking availability of this information. For example, the information on the funding status of a research project is not available via the TISS API.

When considered useful, we expanded the maDMP with additional data fields that have not been specified by the RDA standard. For example, we added identifiers for the research project and the host on which the research data is planned to be stored. The research project identifier is obtained from the TISS API, while the host identifier is retrieved from the re3data API. Also, we added a field containing the license name defined by the EUDAT license selector.

[20] https://www.crossref.org/services/funder-registry/
[21] http://doi.org/10.2312/re3.007

5.2.8 Lifecycle Handling in DMap

The data lifecycle and the role of a DMP are discussed in Section 2.2. As more information becomes available in the phases of a research project, a DMP can contain more detailed and accurate information over time. In DMap, we did not fully implement processes that reflect the temporal shift in the granularity of information contained in a DMP, but we did implement this aspect in some cases.

For example, at an early stage of a research project, the user may not know what type of research data will be created and can indicate this in DMap. Later, the user can specify rough types of data and estimate their volume, or upload files to accurately describe their format and size. However, since we did not fully implement all of the workflows, as stated in Section 5.2.2, a further step to deposit data in a repository or describe already published data is missing.

In summary, it can be said that the processes implemented in DMap correspond more to the requirements of the planning and project phase than the post-project phase.

5.3 TU.it Service Broker

A service broker is a tool to find out which services a provider offers and to obtain them. In the context of data management planning, a researcher wants to find out which ICT services are available at the institution, configure selected services, get a cost estimation for the DMP and have them provided. This workflow is described in Section 3.2.4. The TU.it service broker [Obl20b] is a proof-of-concept implementation of the proposed application service described in the Enterprise Architecture in Chapter 4.

5.3.1 ICT Services

We developed a service broker that provides an interface to the services of TU.it, the local ICT operator at the TU Wien. We selected the following TU.it services for our service broker implementation, which are described in the introduction to the TU Wien case study in Section 5.1.1

- TUfiles
- TUownCloud / TUproCloud
- TUhost

For each of the services we modeled their characteristics such as quality of service descriptions, available configuration and associated costs. For this purpose, we use the specifications described in the next section.

Table 5.8: Services and plans of the TU.it service broker [Obl20b].

Service	Plan	Costs
TUfiles	Standard	paid
TUcloud	TUownCloud	free
	TUproCloud	paid
TUhost	Standard	paid

5.3.2 Implementation of the Open Service Broker API

For the implementation of the TU.it service broker we follow the specifications of the OSBAPI[22]. The OSBAPI is an industry standard for delivering cloud service offerings from different providers to application platforms, as described in Section 2.5. However, the introduction of the service broker concept in data management planning is a novel approach.

The OSBAPI specifies an interface for fetching a service providers' catalog of services with available service plans and provisioning them in a standard manner. This standard interface enables a data management planning application to interact with service brokers for institutional services such as storage, databases or any other computing resources. Hence, the data management planning application can present the user with a catalog of services, including service descriptions, configuration parameters, and associated costs. Depending on the selected service plan and configuration, for example the storage size, the costs can be calculated automatically for the duration the service is needed and stored in the DMP.

Services and Plans

We defined three services in the TU.it service broker - TUfiles, TUcloud and TUhost. Since TUownCloud and TUproCloud are based on the same service, we defined them as two different service plans of TUcloud. While TUownCloud is a free plan with a restricted storage size of 20GB, TUproCloud is a paid service with configurable storage. Table 5.8 depicts a high-level view on the services and plans of the TU.it service broker.

For each of the services and plans, we modeled their characteristics compliant with the OSBAPI specification based on the available information, see Section 5.1.1. For example, we provide links to the documentation of the service and to IT support in the metadata, or provide a list of features in the metadata of a service plan. The costs are also modeled in the metadata of a service plan. Listing 5.2 shows the cost model of the TUhost service implemented in the TU.it service broker.

Listing 5.2: Cost model of the TUhost service implemented in the TU.it service broker [Obl20b].

[22] https://github.com/openservicebrokerapi/servicebroker/blob/v2.13/spec.md

```
                "costs": [
                    {
                        "amount": {
                            "eur": 8
                        },
                        "unit": "vCPU per quarter"
                    },
                    {
                        "amount": {
                            "eur": 8
                        },
                        "unit": "GB RAM per quarter"
                    },
                    {
                        "amount": {
                            "eur": 0.1
                        },
                        "unit": "GB system disk per quarter"
                    },
                    {
                        "amount": {
                            "eur": 0.1
                        },
                        "unit": "GB data disk per quarter"
                    },
                    {
                        "amount": {
                            "eur": 0.25
                        },
                        "unit": "GB high-performance disk per quarter
                            "
                    }
                ],
```

Service Plan Configuration

The OSBAPI specification is flexible with regard to the service configuration parameters. The configuration parameters of a service plan can be defined as properties in a JSON schema[23] object. The JSON schema supports many data types such as array, number, string or boolean in order to express various kinds of configuration parameters. Listing 5.3 shows the JSON schema object for the configuration parameters to create an instance of the TUproCloud service plan. To create an instance of TUproCloud, the service broker requires a string for the cloud storage name and a number for the size in GB.

Listing 5.3: JSON schema object to define the configuration parameters of the TUproCloud service plan in the TU.it service broker. To create a TUproCloud instance, the name and the storage size must be provided [Obl20b].

[23] http://json-schema.org/draft-04/schema#

```
"service_instance": {
    "create": {
        "parameters": {
            "$schema": "http://json-schema.org/draft-04/schema#",
            "properties": {
                "name": {
                    "description": "Unique TU-wide name, not changeable.",
                    "type": "string"
                },
                "size": {
                    "description": "Size of storage in GB.",
                    "type": "int"
                }
            },
            "type": "object"
        }
    }
}
```

5.3.3 Service Broker Interaction

The OSBAPI specifies different types of interaction with the service broker, which are described in the following list.

- **Catalog**: Fetch a catalog of available services and plans.
- **(De)provision**: Create a service instance at the service provider.
- **(Un)bind**: Create a binding to the service instance, for example by returning access details and credentials.
- **Update**: Update the configuration of the service instance, for example the size of a storage, or the number of a VM's virtual CPUs.
- **Last operation**: Return the status of the last operation. This can be used to poll the status of the last requested operation, for example the state of the provisioning.

In the TU.it service broker we did not implement all of the operation endpoints specified by the OSBAPI. As part of this work, we focus on the catalog operation, as it plays the central role in data management planning. The catalog operation serves a data management planning application by providing all the necessary information on available services, plans, configuration and costs which are relevant for creating a DMP. The interaction between the data management planning application and the service broker is described in the sequence diagrams of the proposed Enterprise Architecture in Section 4.6.

In DMap we did not integrate the TU.it service broker as part of this work, but described the intended functionality in the Enterprise Architecture and the graphical mockups, Section 3.3. Figure 5.6 shows a mockup in which the user gets presented with a catalog

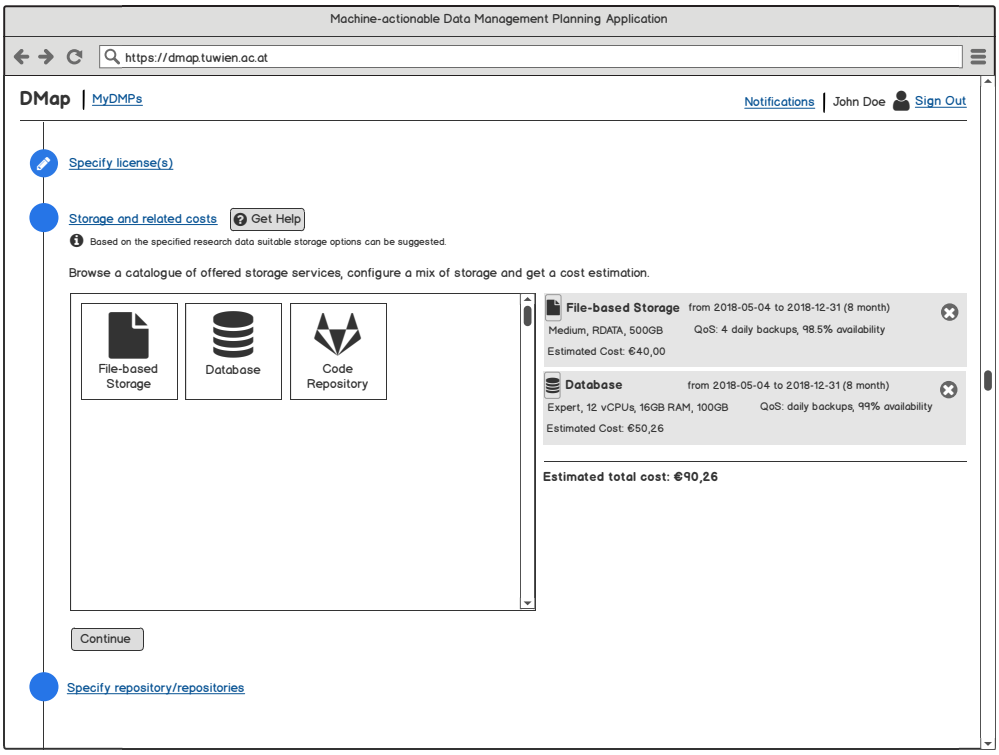


Figure 5.6: Graphical mockup of a data management planning application, as described in Section 3.3. The user can select from a catalog of services provided by the institutional ICT operator such as a file-based storage, database or code repository. Depending on the selected service plans and configuration the costs can be estimated and stored in the DMP. The service plan selection and configuration steps are depicted in the mockups [Obl20a] in Appendix B.

of services provided by the institutional ICT operator, such as a file-based storage, a database or a code repository. The service plan selection and configuration steps are depicted in the mockups in Appendix B on page 113.

Test the Service Broker with Eden

However, we can test the TU.it service broker by interacting with its API. A standard tool for interacting with service brokers which implement the OSBAPI is Eden[24]. With Eden we can request the previously described operations of the service broker. To do this, we need to set the service broker's Uniform Resource Locator (URL) and credentials as environment variables. Listing 5.4 shows the catalog operation requested from the

[24] https://github.com/starkandwayne/eden

TU.it service broker with Eden. Eden lists the available TU.it services and plans with a short description.

Test the Service Broker with cURL

Another way to fetch the service catalog is to make a request to the TU.it service broker via a HTTP client such as cURL[25], as depicted in Listing 5.5. The TU.it service broker returns the service catalog as a JSON payload. Appendix C illustrates an excerpt of the JSON response, describing the TUcloud service with its available service plans TUownCloud and TUproCloud in detail. The JSON response shows that all required information is contained to support our use case and implement a UI as illustrated in Figure 5.6.

Listing 5.4: Fetching the TU.it service catalog with the command line tool Eden [Obl20b].

```
$ eden catalog

Service Name  Plan Name    Description
tu-files      standard     Highly available network drive with standard
                           authorization concept.
tu-host       standard     Configurable, highly available Virtual Machine.
tu-cloud      tu-owncloud  Free file sync and share service for internal use.
~             tu-procloud  File sync and share service for collaboration with
                           external project partners.

4 services
```

Listing 5.5: Request the catalog of services from the TU.it service broker [Obl20b], which is running on the localhost port 8000 with cURL. An excerpt of the JSON response is depicted in Appendix C.

```
curl http://127.0.0.1:8000/v2/catalog -H "X-Broker-Api-Version: 2.13" -u "":""
```

5.3.4 Multiple Service Providers

As described in Section 2.5, an advantage of the OSBAPI is that services from different providers can be used seamlessly in the application platform. The ICT operator can decide which services from which service providers should be made available to the user by registering the respective service brokers in the service catalog of the application platform.

In the Enterprise Architecture, in Chapter 4, we described the Service Catalog Controller application component. Similar to the use in an application platform, this component can be used to register service brokers from different providers. In this way, the ICT operator can offer in-house services together with external services in one catalog by aggregating the services from all registered service brokers. Figure 5.7 illustrates this idea

[25] https://curl.haxx.se/

using the example of TU.it. TU.it can make its current in-house services such as TUfiles or TUhost and future services available through a service broker, but also expand its service portfolio by registering service brokers from external providers such as Amazon Web Service[26] (AWS) or Microsoft Azure[27]. The data management planning application communicates with the Service Catalog Controller, which acts as an intermediate layer for its registered service brokers.

The communication between the application components is also described for some use cases in the process views of the EA in Section 4.6.

Some research infrastructures provide research data services via private cloud powered by OpenStack, such as the University of Edinburgh[28] or the QRIScloud[29] in Australia. Researchers can self-provision cloud services such as virtual machines, Hadoop clusters for big data applications or build and deploy custom workflows for data processing, simulation and analysis. The National eResearch Collaboration Tools and Resources (Nectar) application catalog[30], based on OpenStack Murano[31] offers research tools such as R-Studio, Lime Survey, CKAN, or WordPress for self-provisioning in the cloud. However, the OSBAPI can also provide an interface[32] to Murano applications to make services available that run in the OpenStack cloud infrastructure.

5.4 Summary and Discussion

In this work, we developed workflows based on community ideas, derived a suitable architecture and implemented parts of it in the context of a real example - the TU Wien. The implementation of the proposed maDMP support system gives an insight into how such a system can be realized. It also shows the gaps to the aspirations around maDMPs and the developments needed to overcome them.

With DMap we show that the community ideas, proposed workflows and architectural components can be implemented. We implemented selected parts and achieved the functionality described in the workflows and architecture to a certain degree. We integrated the CRIS of the TU Wien in order to fetch structured information about projects and researchers. We implemented workflows to assist with the specification of research data, license and repository selection. DMap can export a maDMP as JSON that complies with the RDA DMP Common Standard. We implemented a service broker that represents a standard interface to the services provided by TU.it, the local ICT operator. However, we introduced some simplifications of implementation compared to the original

[26] https://aws.amazon.com/partners/servicebroker/
[27] https://osba.sh/
[28] https://www.ed.ac.uk/information-services/research-support/research-computing/ecdf/cloud
[29] https://www.qriscloud.org.au/
[30] https://nectar.org.au/cloud-application-catalogue-released/
[31] https://wiki.openstack.org/wiki/Murano/ApplicationCatalog
[32] https://specs.openstack.org/openstack/murano-specs/specs/liberty/cloud-foundry-service-broker-api.html

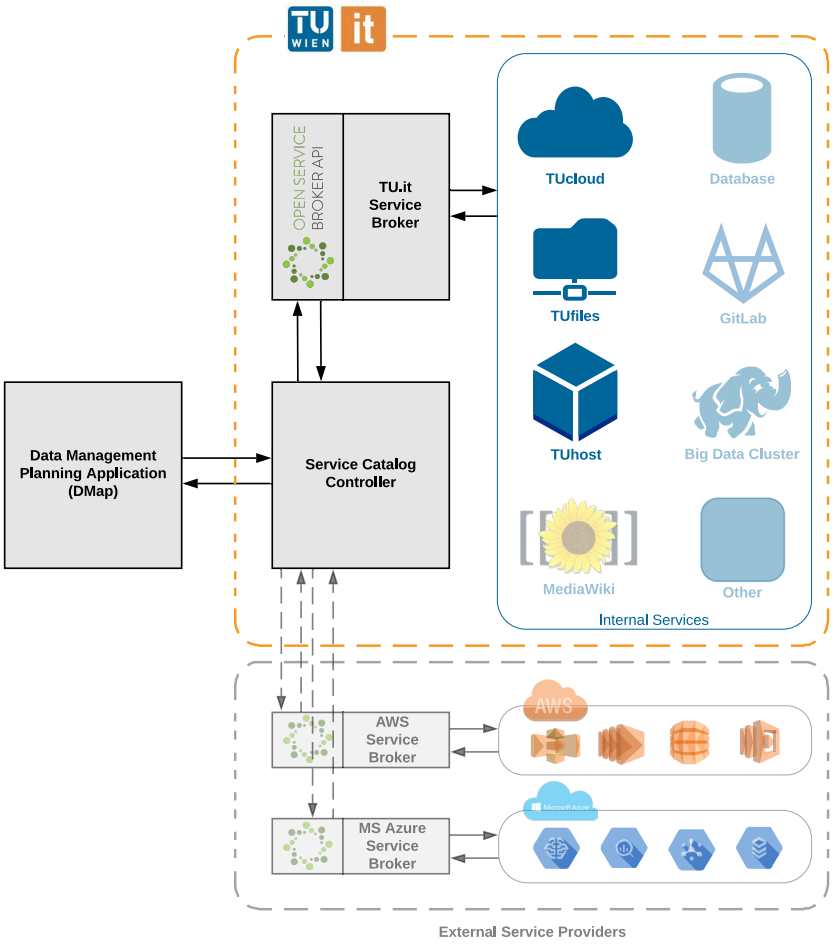


Figure 5.7: TU.it can make its current in-house services such as TUfiles or TUhost and future services available through a service broker, but also expand its service portfolio by registering service brokers from external providers such as Amazon Web Service (AWS) or Microsoft Azure. The data management planning application communicates with the Service Catalog Controller, which acts as an intermediate layer for its registered service brokers.

ideas, workflows and architecture description. For example, we did not model machine-actionable data policy elements, described as one of the ten principles of maDMPs in Section 2.3. We also simplified the proposed repository recommendation service by integrating a repository search in DMap, backed by the repository registry re3data, rather than implementing an actual recommender system. To implement a repository recommendation service as described, relevant maDMP metadata must be aligned with repository metadata, machine-actionable data policies and a suitable recommendation model must be developed.

The implementation of the TU.it service broker shows that the TU.it services can be fully described by implementing the OSBAPI, including configuration parameters and cost models. By implementing the OSBAPI, ICT services can be advertised from within a data management planning application via a standard interface and assist in service configuration and cost estimation.

We integrated several systems into this implementation. However, there are many other systems that can be integrated to implement additional workflows and functionality. For instance, ORCID to have persistent identifiers and additional metadata for researchers, DataCite to fetch metadata about published datasets, the RDA Metadata Standards Directory to find suitable metadata standards, or institutional services such as a ticketing system to get support from DMP experts, or repositories to implement a data flow between a data management planning application and repository systems.

CHAPTER 6

Evaluation

In this chapter we describe how we evaluate the proposed machine-actionable DMP support system. The evaluation is done in two ways - on the one hand we assess the efficiency and on the other hand the effectiveness of the system. The implemented proof-of-concept tool DMap described in Section 5.2 serves as a vehicle for our assessment in order to make statements about the usefulness of the system.

In Section 6.1, we assess the system efficiency by evaluating the processes implemented in DMap. For this purpose, we analyze the extent to which the creation of a DMP in DMap is automated and simplified compared to the conventional way of writing a DMP in free form.

In Section 6.2, we evaluate the machine-actionable DMP created with DMap with regards to its completeness of information that is relevant for the stakeholders. By assessing the completeness of information contained in the maDMP we can make statements about the effectiveness of realizing stakeholder use cases. For this reason, we analyze the requirements of the stakeholder groups ICT operator, repository operator, management and funder and assess how well they are supported by the resulting maDMP. Funders' requirements for DMPs are currently expressed in funder-specific DMP templates. Therefore, we evaluate how well the questions of two major DMP templates can be answered with the information contained in their machine-actionable counterpart.

The two-fold evaluation can also be understood as first evaluating the planning act and then the resulting plan. While the former enables us to draw conclusions about the system efficiency, the latter helps to assess its effectiveness.

6.1 Automation

In order to evaluate the efficiency of the proposed machine-actionable DMP support system, we assess the level of automation and simplification that we achieved with the

Table 6.1: Processes implemented in DMap and their key steps. Implementation details are described in Section 5.2.

Process	Step
I. Start DMP	1.1 Select research project(s)
	1.2 Select people involved in data management
	1.3 Assign contribution role(s)
II. Specify Size and Type	2.1 Group datasets
	2.2 Estimate research data
	2.3 Analyze files
	2.4 Calculate total storage size per dataset
III. Get License	3.1 Select datasets to share
	3.2 Select license(s)
IV. Get Repository	4.1 Select repository/repositories
	4.2 Assign datasets to be deposited

proof-of-concept tool DMap. For this reason, we analyze the processes involved in creating a DMP from the perspective of a researcher who has to write a DMP for a research project. We set the conventional way of writing a DMP in free-form as the baseline for our analysis.

In the analysis we describe the key steps of creating a DMP. For each of the steps, we assess the degree of automation achieved with DMap and describe how and under what assumptions it is more efficient compared to the manual approach. We discuss positive side effects of automation such as the unambiguousness of information and the avoidance of typing errors compared to the manual way of writing a DMP in Section 6.1.3.

6.1.1 Processes and Key Steps

From the processes described as BPMN diagrams in Section 3.2, we derived a suitable Enterprise Architecture, which is described in Chapter 4, and implemented selected processes, as described in Chapter 5. For our analysis, we break down the processes of creating a DMP with DMap into their key steps as illustrated in Table 6.1. However, this set of processes does not claim to be complete, but represents workflows that are implemented in DMap and can be used for evaluation.

The processes we assess include workflows for the research project selection, the selection of contributors and assignment of roles, the specification of research data, the license and repository selection. We do not evaluate the workflow for storage selection and cost estimation because the TU.it service broker was not integrated into DMap as part of this work, but we do discuss its automation potential in Section 6.1.3.

6.1.2 Assessment

In our assessment, we use three categories to annotate the level of automation achieved for each key step of creating a DMP in DMap - manual / semi-automated / fully automated.

For each of the assessed steps, we describe the reasons for the categorization and discuss the differences to the baseline approach of manually writing a DMP. Table 6.2 shows the assessment results.

ad. 1.1 Select research project(s) The selection of a research project in DMap is semi-automatic. Relevant projects are automatically retrieved from the TISS project database and listed (Section 5.2.3). The user can view project details and select one or more projects the DMP is being created for. In this way, project details such as the project ID or the grant ID are safely and automatically imported into the DMP.

ad. 1.2 Select people involved in data management The people involved in data management are selected semi-automatically. Depending on the previously selected research project(s), project members are automatically fetched from the TISS address book and listed in DMap (Section 5.2.3). The user can view person details and select people who are involved in data management. Person details such as their unique ID within TISS or their email address are safely and automatically imported into the DMP.

ad. 1.3 Assign contribution role(s) For the selected people involved in data management, their contribution roles can be selected manually in DMap by choosing from a list of contribution types (Section 5.2.3). Compared to the free-form text baseline, the user can choose from a list of roles defined by a controlled vocabulary, which can simplify the user experience.

ad. 2.1 Group datasets In DMap, research data can be manually grouped into datasets with user-defined names (Section 5.2.4).

ad. 2.2 Estimate research data In DMap, the user can manually estimate the data that will be created in the research project. Therefore, the user can select from a pre-defined list of data types, size ranges, provide a comment and assign the estimation to a dataset (Section 5.2.4). Although the estimation is done manually, choosing from pre-defined values can simplify the user experience compared to describing research data estimations in free-form text.

ad. 2.3 Analyze files Research data can be described semi-automatically in DMap (Section 5.2.4). The user can select files for analysis by dropping them onto the UI drop zone or selecting them in the file explorer. DMap automatically analyses the selected files and displays the identified file formats and sizes in a table. The user can manually provide additional information such as the expected amount of similar files, a comment and the dataset it belongs to.

Table 6.2: Steps of creating a DMP with DMap and their assessed level of automation.

Step	manual	semi-automated	fully automated
1.1 Select research project(s)	☐	☒	☐
1.2 Select people involved in data management	☐	☒	☐
1.3 Assign contribution role(s)	☒	☐	☐
2.1 Group datasets	☒	☐	☐
2.2 Estimate research data	☒	☐	☐
2.3 Analyze files	☐	☒	☐
2.4 Calculate total storage size per dataset	☐	☐	☒
3.1 Select datasets to share	☐	☒	☐
3.2 Select license(s)	☐	☒	☐
4.1 Select repository/repositories	☐	☒	☐
4.2 Assign datasets to be deposited	☒	☐	☐

ad. 2.4 Calculate total storage size per dataset For each of the datasets the total storage size is calculated automatically and displayed in a grid of datasets (Section 5.2.4).

ad. 3.1 Select datasets to share Selecting datasets to share works semi-automatically in DMap. The user is shown a list of previously defined datasets that are marked as "keep closed" by default, and can manually toggle their intended sharing status to "publish" (Section 5.2.5).

ad. 3.2 Select license(s) The selection of a license for the datasets intended for shared use is semi-automatic. A license can be chosen by selecting from a list of 22 common pre-defined licenses or by using the integrated EUDAT license selector which guides the user through a series of questions to narrow down the applicable licenses (Section 5.2.5). URIs that uniquely identify the selected licenses are stored in the DMP.

ad. 4.1 Select repository/repositories In DMap, the user can find and select suitable repositories for depositing the datasets semi-automatically. By having re3data integrated into DMap, the user can apply many filter options and apply a text search to find matching repositories. The user can view the details of the repositories found and select which ones to use Section 5.2.6.

ad. 4.2 Assign datasets to be deposited For each of the selected repositories, the user can manually select which datasets are planned for deposit in the repository and use a date picker to specify how long the datasets should be available Section 5.2.6.

6.1.3 Discussion

The evaluation of the efficiency of processes implemented in DMap shows that most of the processes are semi-automated. Compared to the conventional free-form way of writing a DMP, DMap presents an interactive way of creating a DMP. Instead of being faced with a blank page, the user gets assistance for every step. This is achieved by integrating external information systems (TISS, re3data) and tools (FITS, EUDAT license selector) into DMap, providing a suitable user interface (Section 3.3) for various tasks and offering the user pre-defined options to choose from.

By integrating external services and tools we can simplify the user experience by providing a single UI, instead of switching between different UIs. A positive side effect is that the data from external information systems is imported into the tool in a controlled manner by interacting with the respective service APIs. This enables to unambiguously reference entities such as persons (TISS ID), projects (project ID), grants (grant ID), licenses (Uniform Resource Identifier (URI)) or repositories (re3data ID) in a maDMP. A manual approach to referencing entities is prone to errors, due to possible typing errors.

As discussed in Section 2.3.1, the Horizon 2020 DMP template survey results [GLJ+18] show that users wish for more guidance in terms of recommendations and drop-down options to choose from. We discuss a recommender system for repositories in the workflows (Section 3.2) and the Enterprise Architecture (Chapter 4). However, in DMap we did not implement a repository recommender system, but provided the user with the ability to find suitable repositories by applying filters and text search (Section 5.2.6). Therefore there is even more automation potential than realized in DMap. The license selection can also be further automated as described in Section 3.2.5. DMap provides drop-down options for the assignment of contribution roles, the specification of research data types, sizes and the selection of licenses. We also set reasonable default values to simplify the user experience. For example, the licenses for datasets are set to CC-0 by default, or the project end date is automatically set as the date from when the license should be active. However, further drop-down options can be provided, such as a list of suitable metadata standards to select from (Section 3.2.6).

The selection and automated cost estimation for active data storage (Section 3.2.4) has not been implemented in DMap and therefore could not be evaluated. However, the implementation of the TU.it service broker (Section 5.3) shows that we can implement a standard interface for the interaction with ICT services that enables us to configure ICT services and calculate the costs based on the respective cost models. By integrating service brokers into a data management planning application, we can automate the discovery of suitable ICT services by showing a catalog of services to the user and provide semi-automated assistance in service configuration and cost estimation.

In this evaluation we assessed the level of automation from the perspective of a researcher. However, there are other stakeholders of a maDMP support system who can benefit from automated processes, as discussed in Section 3.1. For example, the ICT operator can get automated support in the management of ICT resources by being integrated into

the data management planning workflow. The funder can semi-automatically evaluate maDMPs because of their machine-actionable format. In Section 3.1 we also describe the automated use cases of the management, repository operators and research support. At the time of writing, these use cases remain aspirational and need to be evaluated after they are implemented.

In the next section, we evaluate the resulting maDMP created with DMap with respect to the completeness of the information that is relevant for the use cases of the stakeholders and their requirements.

6.2 Completeness

In this section, we assess the completeness of the information contained in a maDMP to meet stakeholder requirements. We therefore recall the stakeholder use cases from Section 3.1 and describe the associated requirements for the maDMP. We use the proof-of-concept tool DMap to create a maDMP in the RDA DMP Common Standard v1.0 (Section 5.2.7), which is the subject of our evaluation. For each of the stakeholder requirements we analyze to which extent the resulting maDMP meets the requirement.

Many funders describe their requirements for DMPs in templates. We therefore assess to what extent the questions from the funder DMP templates can be answered with the information contained in the maDMP. For that purpose, we use the DMP templates from two major funders - the Austrian Science Fund (FWF) and the H2020 programme of the European Commission.

6.2.1 MaDMP Example

For the evaluation, we use the proof-of-concept tool DMap to create an example maDMP in the RDA DMP Common Standard v1.0. DMap is not a general tool for data management planning, but tailored to the TU Wien case study. Therefore, the assumptions described in Section 5.1.3 apply.

We describe a fictitious scenario for the creation of the maDMP. The resulting maDMP is not related to the project and people to whom it refers, although the project, funding and contribution information is real, since DMap integrates the TU Wien CRIS and retrieves real information.

In the scenario for creating the example maDMP, we plan the data management of three datasets.

- The first dataset is a collection of satellite images in GeoTIFF format which contains sensitive data because it shows the habitat of an endangered species and therefore shall not be shared.

- The second dataset is the source code of a Python client application which shall be published under the Apache License 2 license and made available on GitHub until ten years after the end of the project.
- The third dataset is a collection of project reports which shall be published under the CC-BY license and citable in a repository that assigns DOIs. The reports shall be available until 15 years after the end of the project.

None of the datasets contains personal information, but ethical issues exist which have been discussed with an ethics committee and documented in a report.

The maDMP describing this scenario is shown in Appendix A on page 109.

Note that we do not describe the plan for managing active data using the services of the local ICT operator as this workflow was not implemented in DMap. However, the RDA specification intends to use multiple instances of a Distribution/Host entity attached to a Dataset entity in order to describe the different locations of the data during all phases. In this way, the plan of storing the data during the active phase somewhere and later depositing the data somewhere else can be expressed by two different Distribution entities with different Host entities.

6.2.2 Stakeholder Requirements

In Section 3.1 we outlined stakeholder use cases of a machine-actionable DMP support system. We recall these use cases and evaluate how well a maDMP can meet the associated requirements. Therefore, we use the previously described example maDMP created in the RDA DMP Common Standard v1.0.

ICT Operator The ICT operators' use cases deal with being informed about computing resources demands in advance and improving the capacity planning based on this information. This requires information about which ICT services and service configurations are planned for the data management of a research project. The maDMP contains information about the host for research data, describing which storage services are being used for data sharing. The host entity includes data fields for describing the quality of service, such as availability or backup type. However, the information on service configuration is not contained in the maDMP. Different ICT services have different configuration parameters, as described in Section 5.1.1. The TU.it service broker (Section 5.3) implements a flexible way to support different configuration parameters for its services. However, this information is not contained in the exported maDMP in the RDA DMP Common Standard, but can be stored with the service broker.

Repository Operator The repository operator's use case is about a controlled deposit of research data into the repository system. Therefore, the repository operator requires information such as the type of data, file formats, volume, used metadata standards, or licensing. All of this information is contained in the maDMP. The type of data is

available in the dataset entity using a controlled vocabulary. File formats are described by their MIME-type or PRONOM PUID, as well as the volume of data in bytes in the distribution entity. Metadata standards can be referenced by URL in the metadata entity. Data licensing information is available in the license entity attached to the distribution entity by using a URI pointing to a specific license. The repository is referenced by its re3data ID in the host entity of the maDMP and is associated with the distribution and dataset entities. In this way it can be clearly mapped which data should be stored in which repository.

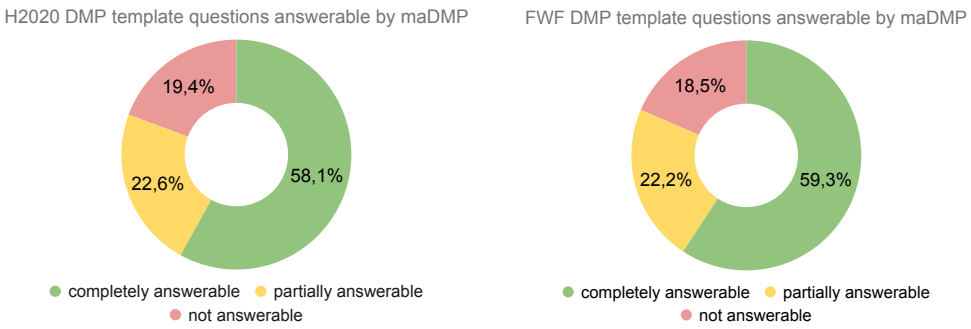
Management The management use case deals with (semi-)automated monitoring and reporting of data management practices at the institution. A DMP database containing the structured information of all maDMPs at the institution can be queried to generate reports on various questions. The question of what kind and amount of research data is being produced at the institution can be answered with the information contained in the maDMP (dataset/type, distribution/format, distribution/byte_size). In order to find out who produces what, the project entity of the maDMP can be used. In our implementation, the maDMP contains a project ID of the TISS project database and can be associated with organizational units or institutes of the TU Wien. The question of how and where research data is being preserved can be answered with the preservation statement of the maDMP and the host entity, which can unambiguously reference a repository by its re3data ID. The question of which licenses are used for data sharing can be answered with a URI of the license contained in the license entity of the maDMP.

Funder The funder use cases are about monitoring and (semi-)automated validation of DMPs. Having a DMP in a machine-actionable format forms an important basis for the DMP review process to be automated. In the next section, we assess the extent to which the maDMP can meet the requirements of two major funding organizations by analyzing how well the questions in their DMP templates can be answered with the information contained in the maDMP.

6.2.3 Funder Templates

As funders are the primary recipients of DMPs, we evaluate how well a maDMP can meet the funders' requirements. For this evaluation, we created a maDMP in the RDA DMP Common Standard v1.0 with our proof-of-concept tool DMap, as described in Section 6.2.1. The maDMP is shown in the Appendix A on page 109. We then use the DMP templates of two major funders, which contain sets of questions and evaluate to what extent they can be answered with the information contained in the maDMP.

The DMP templates of the Austrian Science Fund (FWF) [Aus19] and the European Commission's Horizon 2020 programme [Eur18] are used for this purpose. The FWF template is based on the results of the Science Europe initiative to align the DMP core requirements between European funding organizations [Sci18a].



(a) H2020 DMP template [Eur18]. (b) FWF DMP template [Aus19].

Figure 6.1: Analysis results for both DMP templates. More than 80% of the questions of both DMP templates can be answered completely or partially with the information contained in the maDMP.

For each of the 31 questions in the H2020 template and 27 questions in the FWF template, we assess whether the question can be answered completely/partially/not with the information contained in the maDMP. If the question can be answered completely or partially we list the corresponding maDMP fields. The result of this exercise is shown in Appendix E.

We implemented selected processes in DMap (Section 5.2.2) and therefore do not use all data fields specified in the RDA DMP Common Standard in the exported maDMP (Section 5.2.7). However, we also take into account the data fields not used in DMap for the evaluation and mark them in red, as shown in Appendix E.

The analysis shows similar results for both DMP templates, although the questions differ in topic and granularity. More than half of the questions (58,1% H2020, 59,3% FWF) of both templates can be completely answered with the information contained in the maDMP. About one fifth to one quarter (22,6% H2020, 22,2% FWF) of questions from both templates can be answered partially, and about a fifth of the questions (19,4% H2020, 18,5% FWF) cannot be answered. Figure 6.1 depicts the analysis results for both DMP templates.

We further analyze the questions of the DMP templates based on their question categories, whereby both DMP templates have their own question categories.

H2020 DMP Question Categories Table 6.3 shows the analysis results of the answerability of H2020 DMP template questions grouped by their categories.

The analysis shows that more than half of the questions in the category Data summary can be completely answered, while the remaining questions of this category cannot be answered. These answerable questions are related to the data type, format, size, origin and reuse of data, and can be answered by the information collected in the maDMP,

Table 6.3: Results of the evaluation exercise, showing the degree to which the 31 questions of the Horizon 2020 DMP template [Eur18] can be answered by the information contained in the maDMP (Appendix E, page 119).

H2020 DMP question category		Question answerable		
		completely	partially	not
Data summary		57,1% (4)	0,0% (0)	42,9% (3)
FAIR data	Findable data	50,0% (3)	33,3% (2)	16,7% (1)
	Accessible data	60,0% (3)	40,0% (2)	0,0% (0)
	Interoperable data	50,0% (1)	0,0% (0)	50,0% (1)
	Reusable data	60,0% (3)	40,0% (2)	0,0% (0)
Allocation of resources		66,7% (2)	33,3% (1)	0,0% (0)
Data security		100,0% (1)	0,0% (0)	0,0% (0)
Ethical aspects		100,0% (1)	0,0% (0)	0,0% (0)
Other		0,0% (0)	0,0% (0)	100,0% (1)
		58,1% (18)	**22,6% (7)**	**19,4% (6)**

such as data types, formats (MIME-type, PRONOM PUID), byte sizes, or DOIs. The questions that cannot be answered by the maDMP are open questions that require an explanatory text, such as "Explain the relation to the objectives of the project".

The questions related to the FAIRness of data can mainly be answered completely or partially, whereby its subset of questions related to the data interoperability can only by answered by 50%.

Two thirds of the questions about the allocation of resources can be answered completely, and one third partially. The maDMP supports descriptive text for each cost item and therefore these questions can be answered. The question about the clear responsibilities for data management can only be answered partially, because the maDMP contains a data field for the role but does not describe clear responsibilities.

The question of data security can be answered completely because the maDMP contains a free-form text field and the data security measures are not divided into a specific domain terminology. The same applies for other questions, such as the question about data quality assurance in the category of data reusability.

FWF DMP Question Categories The analysis results of the answerability of questions from the FWF DMP template, grouped by their categories, are listed in Table 6.4.

The analysis shows that the questions in the category "Description of the data" can be answered completely by 25% and partially by another 25%. Compared to the H2020 DMP template, the FWF DMP template has more coarse-grained questions. The type, format and size of data is asked in a single question in the FWF DMP template, while the Horizon 2020 DMP template asks two questions for the same information. Again,

Table 6.4: Results of the evaluation exercise, showing the degree to which the 27 questions of the FWF DMP template [Aus19] can be answered by the information contained in the maDMP (Appendix E, page 119).

FWF DMP question category	Question answerable		
	completely	partially	not
Data Officer	100,0% (1)	0,0% (0)	0,0% (0)
I.1 Description of the data	25,0% (1)	25,0% (1)	50,0% (2)
II. 1 Metadata standards	100,0% (1)	0,0% (0)	0,0% (0)
II.2 Documentation of data	0,0% (0)	33,3% (1)	66,7% (2)
II.3 Data quality control	100,0% (2)	0,0% (0)	0,0% (0)
III.1 Data sharing strategy	100,0% (3)	0,0% (0)	0,0% (0)
III.2 Data storage strategy	71,4% (5)	28,6% (2)	0,0% (0)
IV.1 Legal aspects	25,0% (1)	50,0% (2)	25,0% (1)
IV.2 Ethical aspects	100,0% (2)	0,0% (0)	0,0% (0)
	59,3% (16)	**22,2% (6)**	**18,5% (5)**

in addition to the description of data type, format and size, open questions requiring explanatory text are asked which are not covered by data fields of the maDMP.

The questions about the data officer, used metadata standards, data quality control, data sharing strategy and ethical aspects can be answered completely with the information from the maDMP.

None of the questions related to the documentation of data can be answered completely. Again, one reason for this is the coarse granularity of the questions, which makes it difficult to decide whether a question can be answered completely or not, and was therefore evaluated as partially answerable if some maDMP data fields are suitable.

About 70% of the data storage strategy questions can be answered completely, and the rest partially.

One quarter only of the questions related to legal aspects can be answered completely. A specific license can be referenced by its URI to attach to the data. However, the maDMP does not provide another data field to clarify data ownership and to explain the reasons for restrictions on data use rights if these are restricted.

6.2.4 Discussion

The evaluation of whether the created maDMP meets the stakeholders' requirements shows that the maDMP does not contain all information that is relevant for realizing the use cases of the stakeholders. In case of the ICT operator, the information of what type of data and volume will be created is available in the maDMP, but not which ICT services and configurations are planned to be used. So this information is not available in the maDMP but can be stored somewhere else. However, the service broker (Section 5.3)

acting as an intermediary layer between the DMP application and the service provider can store this information and make it available for the ICT operators' capacity planning.

The maDMP can meet the requirements of the repository operator describing data and metadata. In case of the institutional management, it depends on which specific questions they are interested in. As we have shown some questions can be answered with the information in the maDMP. However, having the DMPs in a machine-actionable format and queryable from a DMP database enables to explore the data management practices of researchers at the institution in a number of ways.

The evaluation of whether the created maDMP meets the funders' requirements shows that more than half of the questions in the DMP templates of two major funding organizations can be answered completely. Of these questions, many can be answered with the type-safe, machine-actionable information in the maDMP, such as PIDs, URIs, numerical values or data fields with a controlled vocabulary, while others can be answered with the free-form text fields of the maDMP.

However, the remaining questions of the DMP templates cannot be answered or only partially with the information contained in the maDMP. One reason for this is that some of these questions require explanations or descriptions for which the maDMP does not provide suitable data fields or the information in the maDMP is not meaningful enough to adequately answer the questions. This shows that the maDMP in its current representation cannot contain all information required by funders.

This is not surprising since a DMP is expected to still contain human-readable narratives [SJMM17]. The maDMP in the RDA DMP Common Standard provides machine-actionable data fields but also free-form text fields. While machine-actionable data fields are more explicit, free-form text fields are more expressive and make it possible to articulate complexity in natural language.

If the funder requirements for DMPs do not converge, it is difficult to design a data model that supports all kinds of questions from different DMP templates. On the one hand, there are developments to harmonize the requirements for DMPs among the funding organizations by establishing core requirements for DMPs [Sci18b]. On the other hand, data management practice can vary dramatically between disciplines and communities. The Domain Data Protocol (DDP) is a framework that allows communities to define RDM guidelines and procedures meeting discipline-specific needs, while including minimal conditions such as following the FAIR principles [Sci18b]. DDPs are to be used as DMP templates, with only minimal input being required. Both developments show that there is a desire to unify the DMP landscape without neglecting community and discipline-specific needs.

Therefore, maDMPs in their current representation do not fully replace conventional DMPs, but augment them with machine-actionability.

CHAPTER 7

Conclusions and Outlook

DMPs are the only tool that comprehensively addresses RDM and covers the entire lifecycle of research data. Machine-actionable DMPs are the next step in the evolution and accommodate the potential to integrate RDM systems from all phases of the research data lifecycle in order to establish a natural transition trough all phases. They further make RDM more transparent and traceable for humans and machines.

Research institutions can build their research data infrastructure around maDMPs to bring researchers together with departments and services. By automating tasks for data management planning, researchers can be supported on their journey to good data management practice. The machine-actionable DMP information collected on the way helps streamline data management workflows that involve multiple stakeholders and systems.

In this thesis, we explored use cases of machine-actionable DMPs in the context of a research institution or university with its systems and services and proposed an architecture for a maDMP support system. A proof-of-concept implementation shows that the use cases are feasible and certain tasks can be automated. However, as indicated in the methods Section 1.4 of the introductory Chapter 1, maDMPs are topic of ongoing research and multiple iterations may be required to bring the maDMP vision to practice.

7.1 Research Questions Revisited

We have defined a set of research questions for this work in Section 1.3. In this section, we revisit these questions and discuss how they were answered.

RQ1 What are the requirements for a machine-actionable data management planning support system in an institutional context?

We collected requirements on three levels - business logic (Section 3.2), UI (Section 3.3) and architecture (Section 4.3). The requirements for the business logic were defined by the community and expressed as business processes using BPMN. Requirements for the UI were collected with the help of graphical mockups and the feedback from stakeholders. By relating the community use cases to an institutional data policy we could identify drivers for change in an institutional research data infrastructure and assess them to derive architectural goals and associated requirements and constraints.

(a) What are the workflows (tasks) of machine-actionable data management planning?

We defined nine workflows (Section 3.2) for tasks of data management planning. The workflows implement maDMP use cases and involve the researcher as a central actor and institutional stakeholders and services. They cover different DMP themes and different phases of the research data lifecycle and do not claim to be complete.

RQ2 What is a suitable architecture supporting machine-actionable data management planning at a research institution?

(a) How can the requirements be mapped into a modular architecture?

In the EA (Chapter 4) we integrated the workflows (Section 3.2) into the business process layer of the architecture and considered requirements associated with architectural goals (Section 4.3). From the processes and sub-processes of the business layer we derived application services that can implement the former. Application services were further dissected into application components. By arranging architectural components around business capabilities, we achieved a modular architecture that is polyglot and enables self-contained components to be independent of programming language and technology.

(b) What services are needed?

In the EA we propose thirteen services (Table 4.3) that are required to implement the business logic. The services vary in complexity and form. Some services are intended to provide a GUI, while others offer an API.

(c) Which services are institution-specific and which could be outsourced and shared with other institutions?

Most of the services are independent from the institution. However, institutions have different systems and services in place. Therefore, the services that interact with institutional systems such as the CRIS need to be implemented

institution-specific. However, standard systems can be supported. The ICT services and computing resources offered at institutions are abstracted by the *Service Broker*, whose standard interface must be implemented institution-specific. The implementation of the *Metadata Standard Discovery* may vary depending on the research area of the institution.

The architecture is designed to be deployed on an institutional level. However, there are some services that can be outsourced and shared between institutions or the public. These services include the *Repository Recommender*, *Metadata Standard Discovery*, *File Characterizer*, *Metadata Importer*, *Repository Ingestor*, *Help Desk*.

(d) How can we integrate the research data management services offered at the institution into data management planning?

We propose the implementation of a service broker that follows the OSBAPI (Section 2.5), an open standard for cloud service provisioning. RDM services such as file storage, databases, code repositories, VMs or research tools can be integrated into data management planning with the help of a service broker. The service broker advertises the available services to the researcher in a catalog, where services can be selected and configured. Based on the service configuration and the cost model, the service cost can be estimated.

The implementation of the TU.it service broker (Section 5.3) shows that the ICT services offered at the TU Wien can be fully described by following the specification of the OSBAPI, including service configuration parameters and costs.

RQ3 To what extent does the proposed system make the DMP process more efficient?

(a) Which tasks of data management planning can be supported with system integration and automation?

We defined nine BPMN workflows (Section 3.2) for tasks of data management planning. Each workflow is designed to improve the efficiency of data management planning by means of automation and system integration. We implemented four of the workflows in the proof-of-concept implementation DMap (Chapter 5). The implemented workflows automate the selection of research projects and people by integrating TISS, the specification of research data supported by FITS, the license selection by integrating the EUDAT license selector and the repository selection by integrating re3data.

We implemented the TU.it service broker which is the central service to automate the storage selection and cost estimation. However, we did not integrate it into DMap as part of this work.

(b) Which degree of automation can we achieve for data management planning tasks?

In Section 6.1 of the evaluation Chapter 6, we assessed the level of automation achieved for DMP tasks in the proof-of-concept implementation DMap. The evaluation shows that most of the tasks can be semi-automated. However, in DMap we introduced some simplifications of the tasks compared to the defined BPMN workflows, and therefore there is even more potential for automation than currently realized.

RQ4 To what extent do the resulting DMPs meet the stakeholder requirements?

As a result of the data management planning process with DMap, we exported a DMP in the RDA DMP Common Standard (Section 5.2.7). We evaluated how well the RDA DMP meets the requirements of the ICT operator, repository operator, management and funder (Section 6.2.2). The evaluation shows that the DMP fulfills the requirements of the repository operator and management well for some use cases. The requirements for the ICT operator however are not met. The information about which services including configuration are planned to be used cannot be sufficiently recorded in the DMP, but can be captured with the service broker.

(a) To what extent can they follow the DMP templates of major funding organizations?

We evaluated how well the exported RDA DMP can follow the DMP templates of the FWF and the EC's H2020 programme (Section 6.2.3). The evaluation shows similar results for both DMP templates. More than half of the questions asked in both templates can be fully answered with the information contained in the maDMP. However, the remaining questions cannot be answered or only partially. One reason for this is that some of these questions require explanations or descriptions for which the maDMP does not provide suitable data fields or the information in the maDMP is not meaningful enough to adequately answer the questions. MaDMPs in their current representation therefore cannot contain all information required by funders and do not fully replace conventional DMPs, but augment them with machine-actionability.

7.2 Limitations and Future Work

The proposed maDMP support system provides support in data management cost estimation. It estimates the cost of ICT services used in RDM based on the service configuration and the cost model defined in the service broker. However, there are many

other costs involved in RDM such as the cost of staff needed for data management, repository fees, preservation cost, etc. which we do not consider in the current design. For effective data management planning, however, a DMP tool may offer comprehensive support for costing RDM.

Machine-actionable data policies should express elements of research funders' and institutions' data policies to help automate data management (planning) tasks in compliance with their policies. Some of the automated workflows presented in this work use the principle of machine-actionable data policies. However, machine-actionable data policies are still an open problem. To fully exploit the automation potential described, machine-actionable data policies need to be developed.

Machine-actionable DMPs are still in their infancy and more research is required. In this work, we focused on providing automated support for the creation of maDMPs and enabling related use cases of institutional stakeholders. However, we did not elaborate on the use cases of the research funder, including DMP monitoring and automated support for reviewing DMPs. Future work may include further investigations into community use cases, especially in the domain of recommender systems. Repository recommendation using maDMPs and machine-learning techniques is a promising field of research.

APPENDIX A

RDA DMP Common Standard JSON

Listing A.1: Example maDMP in RDA DMP Common Standard v1.0 [MWN19].

```
{
  "dmp": {
    "title": "DMP for our new project",
    "dmp_id": {
      "identifier": "89bf3739-a352-4b70-9ef3-3322f2ce6b51",
      "type": "other"
    },
    "description": "This DMP is for our new project.",
    "language": "eng",
    "created": "2020-01-19T12:40:17.303Z",
    "modified": "2020-01-19T12:40:17.303Z",
    "contact": {
      "name": "Alexandra von Beringe",
      "mbox": "alexandra.beringe@tuwien.ac.at",
      "contact_id": {
        "identifier": "https://www.tiss.tuwien.ac.at/person/54309",
        "type": "other"
      }
    },
    "contributor": [
      {
        "name": "Alexandra von Beringe",
        "mbox": "alexandra.beringe@tuwien.ac.at",
        "contributor_id": {
          "identifier": "https://www.tiss.tuwien.ac.at/person/54309",
          "type": "other"
        },
        "role": [
          "ContactPerson",
          "DataManager",
          "DataCurator"
        ]
      },
      {
        "name": "Matthias Schramm",
        "mbox": "matthias.schramm@tuwien.ac.at",
        "contributor_id": {
          "identifier": "https://www.tiss.tuwien.ac.at/person/305962",
          "type": "other"
        },
```

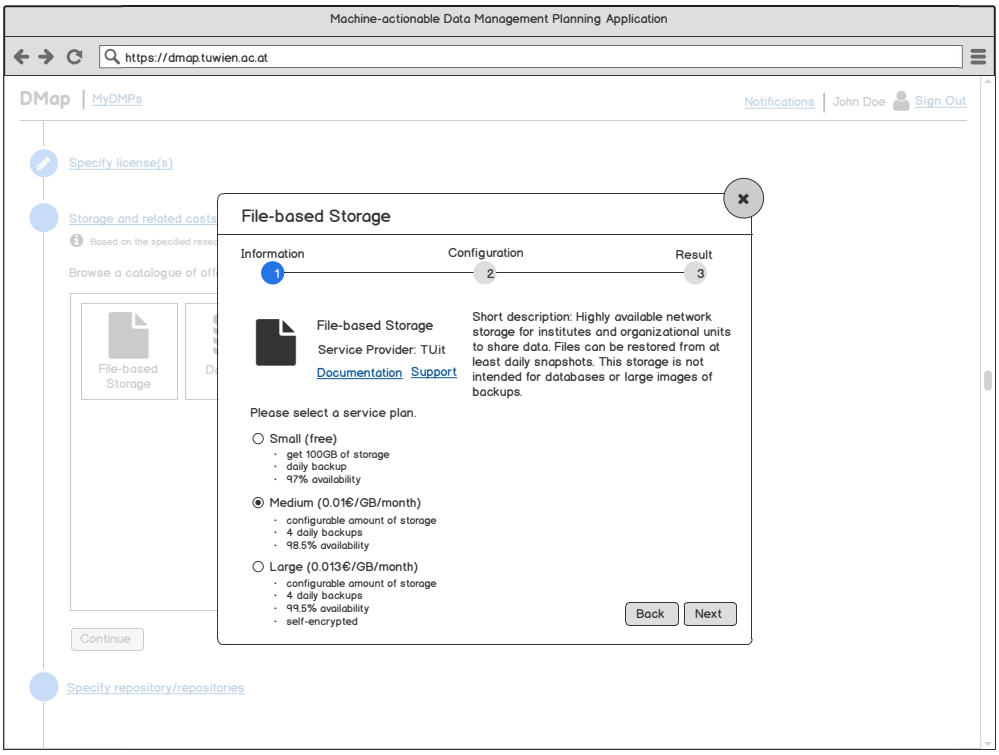
```
        "role": [
          "ProjectManager"
        ]
      }
    ],
    "ethical_issues_exist": "yes",
    "ethical_issues_report": "https://docs.google.com/document/d/17yos96EoNP3eimfyQo7rBBu3FE9t1VzMr-YzS7x48a8/edit?usp=sharing",
    "ethical_issues_description": "There are ethical issues, because...",
    "project": [
      {
        "title": "openEO - a common, open source interface between Earth Observation data infrastructures and front-end applications",
        "project_id": {
          "identifier": "https://tiss.tuwien.ac.at/api/pdb/rest/project/v3/1428966",
          "type": "tiss-pdb"
        },
        "start": "2017-10-01",
        "end": "2020-09-30",
        "description": "<p>The capabilities of the latest generation of Earth observation satellites to collect large volumes of diverse and thematically rich data are unprecedented. For exploiting these valuable data sets, many research and industry groups have started to shift their processing into the cloud. Although the functionalities of existing cloud computing solutions largely overlap, there are all custom-made and tailored to the specific data infrastructures. This lack of standards not only makes it hard for end users and application developers to develop generic front-ends, but also to compare the cloud offerings by running the same analysis against different cloud back-ends. To solve this, a common interface that allows end- and intermediate users to query cloud-based back offices and carry out computations on them in a simple way is needed. The openEO project will design such an interface, implement it as an open source community project, bind it to generic analytics front-ends and evaluate it against a set of relevant Earth observation cloud back offices. The openEO interface will consist of three layers of Application Programming Interfaces, namely a core API for finding, accessing, and processing large datasets, a driver APIs to connect to back offices operated by European and worldwide industry, and client APIs for analysing these datasets using R, Python and JavaScript. To demonstrate the capability of the openEO interface, four use cases based chiefly on Sentinel-1 and Sentinel-2 time series will be implemented. openEO will simplify the use of cloud-based processing engines, allow switching between cloud-based back office providers and comparing them, and enable reproducible, open Earth observation science. Thereby, openEO reduces the entry barriers for the adaptation of cloud computing technologies by a broad user community and paves the way for the federation of infrastructure capabilities.</p><p> </p>",
        "funding": [
          {
            "funder_id": {
              "identifier": "European Commission - Framework Programme",
              "type": "other"
            },
            "grant_id": {
              "identifier": "EO-2-2017",
              "type": "other"
            }
          }
        ]
      }
    ],
    "dataset": [
      {
        "title": "Satellite Images",
        "dataset_id": {
          "identifier": "",
          "type": "other"
        },
        "personal_data": "no",
        "sensitive_data": "yes",
        "type": [
          {
            "name": "other",
            "description": "Images from satellite x of region y during period z"
          }
        ],
        "distribution": [
```

```
        {
          "title": "Planned distribution",
          "format": [
            "image/tiff"
          ],
          "byte_size": 27855370000,
          "data_access": "closed"
        }
      ]
    },
    {
      "title": "Project Reports",
      "dataset_id": {
        "identifier": "",
        "type": "other"
      },
      "personal_data": "no",
      "sensitive_data": "no",
      "type": [
        {
          "name": "Standard office documents",
          "description": "Documents for reporting the project progress"
        }
      ],
      "distribution": [
        {
          "title": "Planned distribution",
          "format": [],
          "byte_size": 100000000,
          "data_access": "open",
          "license": [
            {
              "name": "Creative Commons Attribution (CC-BY)",
              "license_ref": "http://creativecommons.org/licenses/by/4.0/",
              "start_date": "2018-02-28"
            }
          ],
          "host": {
            "title": "Zenodo",
            "host_id": {
              "identifier": "https://www.re3data.org/repository/r3d100010468",
              "type": "re3data"
            },
            "url": "https://zenodo.org/",
            "description": "ZENODO builds and operates a simple and innovative service that
                enables researchers, scientists, EU projects and institutions to share and
                showcase multidisciplinary research results (data and publications) that are not
                 part of the existing institutional or subject-based repositories of the
                research communities.\nZENODO enables researchers, scientists, EU projects and
                institutions to:\neasily share the long tail of small research results in a wide
                 variety of formats including text, spreadsheets, audio, video, and images
                across all fields of science.\ndisplay their research results and get credited
                by making the research results citable and integrate them into existing
                reporting lines to funding agencies like the European Commission.\neasily access
                 and reuse shared research results.",
            "support_versioning": "unknown",
            "storage_type": "repository",
            "pid_system": [
              "doi"
            ]
          },
          "available_until": "2035-09-30"
        }
      ]
    },
    {
      "title": "Client Application",
      "dataset_id": {
        "identifier": "",
        "type": "other"
      },
      "personal_data": "no",
      "sensitive_data": "no",
```

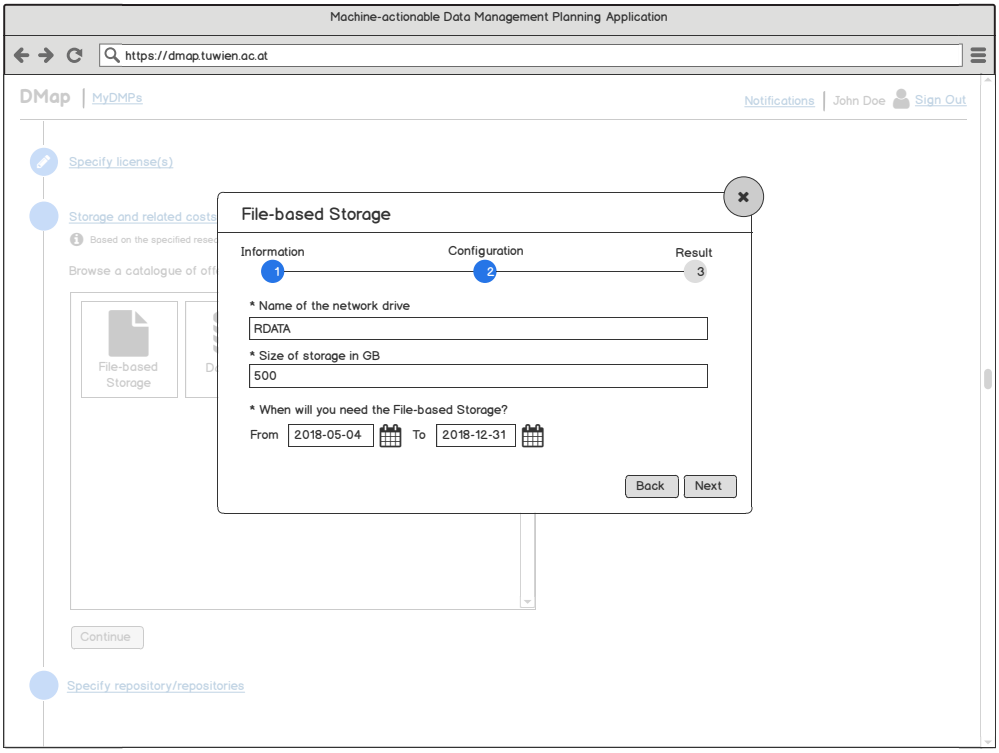
```
        "type": [
          {
            "name": "Source code",
            "description": "Python client for interacting with the API"
          }
        ],
        "distribution": [
          {
            "title": "Planned distribution",
            "format": [],
            "byte_size": 1000000000,
            "data_access": "open",
            "license": [
              {
                "name": "Apache License 2",
                "license_ref": "http://www.apache.org/licenses/LICENSE-2.0",
                "start_date": "2018-01-01"
              }
            ],
            "host": {
              "title": "GitHub",
              "host_id": {
                "identifier": "https://www.re3data.org/repository/r3d100010375",
                "type": "re3data"
              },
              "url": "https://github.com",
              "description": "GitHub is the best place to share code with friends, co-workers,
                  classmates, and complete strangers. Over three million people use GitHub to
                  build amazing things together. With the collaborative features of GitHub.com,
                  our desktop and mobile apps, and GitHub Enterprise, it has never been easier for
                   individuals and teams to write better code, faster. Originally founded by Tom
                  Preston-Werner, Chris Wanstrath, and PJ Hyett to simplify sharing code, GitHub
                  has grown into the largest code host in the world.",
              "support_versioning": "yes",
              "storage_type": "repository",
              "pid_system": [
                "none"
              ]
            },
            "available_until": "2030-09-30"
          }
        ]
      }
    ]
  }
}
```

APPENDIX B

DMap Selected Mockups



(a) Selection of a service plan with different qualities of service and costs.



(b) Configuration of the service parameters such as the name and size of the network drive.

Figure B.1: Graphical mockups of a data management planning application, see Section 3.3. The user can choose a service plan (B.1a) and configure the service (B.1b) [Obl20a].

APPENDIX C

TU.it Service Broker JSON

Listing C.1: Excerpt from the response of the TU.it service broker, showing the service plans TUownCloud and TUproCloud available in its service catalog [Obl20b].

```
{
    "bindable": true,
    "description": "File Sync and Share service located on servers of the TU Wien.",
    "id": "c4573c4b-0fec-4c2d-b650-a3daa91a3bf0",
    "metadata": {
        "displayName": "TUCloud",
        "documentationUrl": "https://www.it.tuwien.ac.at/tuprocloud/",
        "imageUrl": "http://example.com/tucloud_logo.png",
        "longDescription": "With our File Sync and Share service, you can save your data to
             a \"virtual memory stick\". The open source software ownCloud runs on TU.it
            servers and basically offers the familiar features also provided by public
            cloud systems such as Dropbox.",
        "providerDisplayName": "TU.it",
        "supportUrl": "https://support.tuwien.ac.at/assystnet/"
    },
    "name": "tu-cloud",
    "plan_updateable": true,
    "plans": [
        {
            "description": "Free file sync and share service for internal use.",
            "free": true,
            "id": "b1e8a5fd-0abc-4259-8d66-ab6fe8ee8b1d",
            "metadata": {
                "bullets": [
                    "20 GB of personal storage space",
                    "Running on TU.it servers - your data is present locally on our systems
                        ",
                    "Data access possible via clients, web and WebDAV",
                    "Synchronisation with any number of devices is either automatic or in
                        accordance with settings you make yourself"
                ],
                "displayName": "TUownCloud"
            },
            "name": "tu-owncloud"
        },
        {
            "description": "File sync and share service for collaboration with external
                project partners.",
            "free": false,
            "id": "433b4d74-6ed9-41f5-81e6-4bef7fd66c1f",
            "metadata": {
                "bullets": [
                    "Collaborate with external project partners",
```

```
                        "File-synchronization and sharing",
                        "Configure access / authorization of members",
                        "Several 100GB of storage possible"
                    ],
                    "costs": [
                        {
                            "amount": {
                                "eur": 0.03
                            },
                            "unit": "GB per quarter"
                        }
                    ],
                    "displayName": "TUproCloud"
                },
                "name": "tu-procloud",
                "schemas": {
                    "service_binding": {
                        "create": {
                            "parameters": {
                                "$schema": "http://json-schema.org/draft-04/schema#",
                                "properties": {
                                    "xxxx": {
                                        "description": "Some parameter needed for binding the
                                            service instance.",
                                        "type": "string"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    },
                    "service_instance": {
                        "create": {
                            "parameters": {
                                "$schema": "http://json-schema.org/draft-04/schema#",
                                "properties": {
                                    "name": {
                                        "description": "Unique TU-wide name, not changeable.",
                                        "type": "string"
                                    },
                                    "size": {
                                        "description": "Size of storage in GB.",
                                        "type": "int"
                                    }
                                },
                                "type": "object"
                            }
                        },
                        "update": {
                            "parameters": {
                                "$schema": "http://json-schema.org/draft-04/schema#",
                                "properties": {
                                    "size": {
                                        "description": "Size of storage in GB.",
                                        "type": "int"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                }
            ],
            "tags": [
                "cloud-storage",
                "file-sync",
                "share"
            ]
        },
```

APPENDIX D

EA Layered View

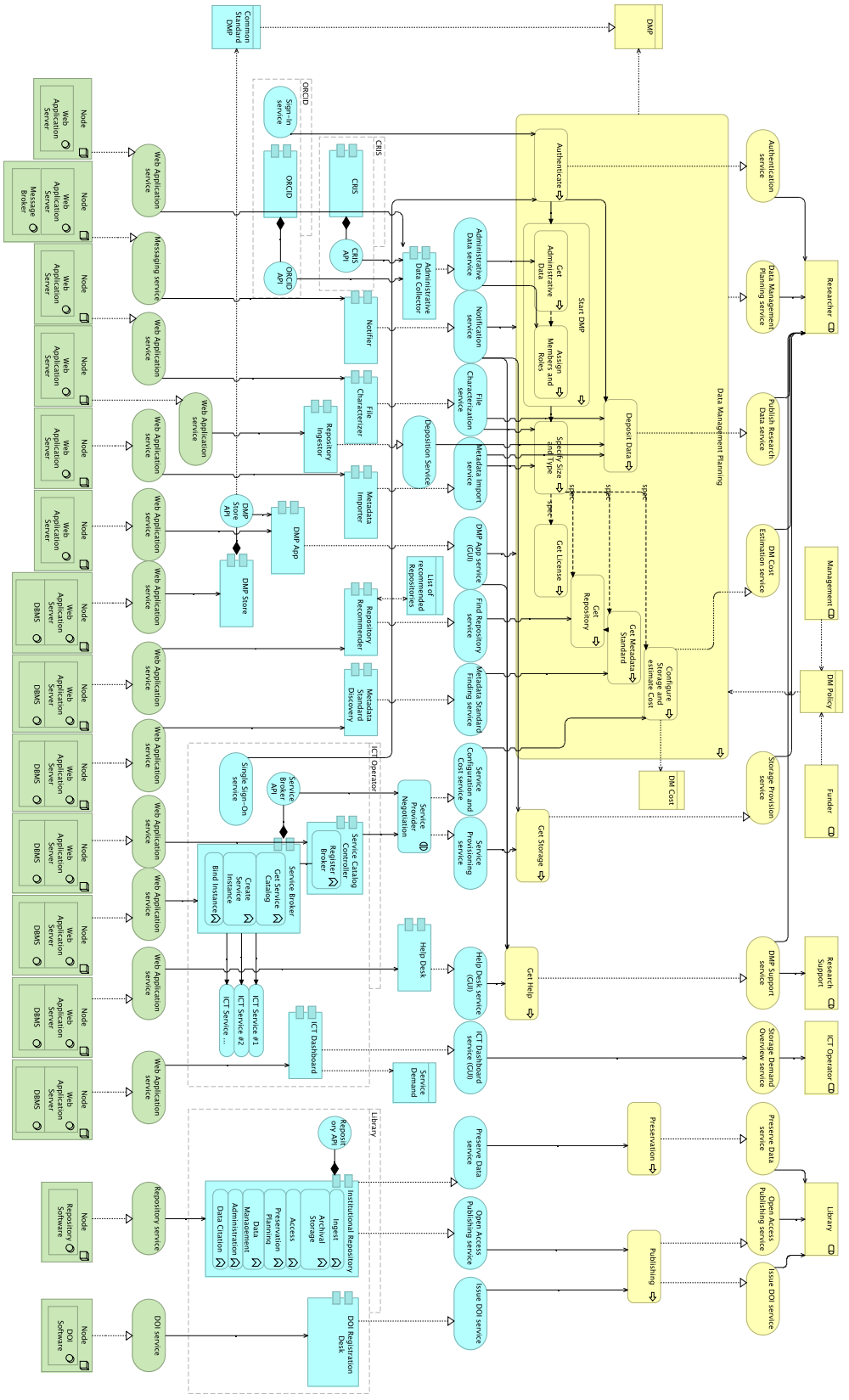


Figure D.1: Enterprise Architecture layered view - business (yellow), application (blue), technology (green).

APPENDIX E

Funder DMP Templates Evaluation

H2020 DMP template questions		Question answerable: completely	partially	not	Corresponding maDMP fields	Comment
Data summary	State the purpose of the data collection/generation			x	–	
	Explain the relation to the objectives of the project			x	–	
	Specify the types and formats of data generated/collected	x			dataset/type, distribution/format	Controlled vocabulary for data types, data format expressed as MIME-type or PRONOM PUID
	Specify if existing data is being re-used (if any)	x			dmp/dataset/*	Reference and import metadata from existing dataset, e.g. via DataCite DOI if available or provide dataset description
	Specify the origin of the data	x			dataset/dataset_id, distribution/access_url, dataset/description	E.g. DOI if available or provide dataset description
	State the expected size of the data (if known)	x			distribution/byte_size	Size in bytes
	Outline the data utility: to whom will it be useful			x	–	
FAIR data	Outline the discoverability of data (metadata provision)	x			dataset/dataset_id, metadata/description	E.g. DOI registered with DataCite metadata. Textual description of metadata for discoverability.
	Outline the identifiability of data and refer to standard identification mechanism. Do you make use of persistent and unique identifiers such as Digital Object Identifiers?	x			dataset/dataset_id, host/pid_system	Dataset can be referenced by PID. Description of what type of PID systems the host supports.
	Outline naming conventions used			x	–	
	Outline the approach towards search keyword		x		dataset/keyword	List of keywords related to the dataset only, but description missing
	Outline the approach for clear versioning		x		host/support_versioning	Only describes the capability of the host to support versioning, but not how versioning will be applied to the data
	Specify standards for metadata creation (if any). If there are no standards in your discipline describe what metadata will be created and how	x			metadata/metadata_standard_id, metadata/description	Metadata standards can be referenced by an URL + free-form text description possible
	Specify which data will be made openly available? If some data is kept closed provide rationale for doing so		x		distribution/data_access	Flags dataset access as {open, shared, closed}, but rational text is missing
	Specify how the data will be made available	x			distribution/*, distribution/host	Location and data host can be specified.
	Specify what methods or software tools are needed to access the data? Is documentation about the software needed to access the data included? Is it possible to include the relevant software (e.g. in open source code)?	x			dataset/technical_resource	Description of technical resources for accessing and processing the data.
	Specify where the data and associated metadata, documentation and code are deposited	x			distribution/host/*	Data host can be specified.
	Specify how access will be provided in case there are any restrictions		x		distribution/data_access, distribution/access_url	Access URL can be provided for datasets with restricted access, but a description cannot be provided.
	Assess the interoperability of your data. Specify what data and metadata vocabularies, standards or methodologies you will follow to facilitate interoperability.	x			dataset/metadata/*	Metadata for interoperability can be referenced or described.
	Specify whether you will be using standard vocabulary for all data types present in your data set, to allow inter-disciplinary interoperability? If not, will you provide mapping to more commonly used ontologies?			x	–	
	Specify how the data will be licenced to permit the widest reuse possible	x			license/license_ref	Licenses can be referenced by URI.
	Specify when the data will be made available for re-use. If applicable, specify why and for what period a data embargo is needed		x		license/start_date	Embargo can be specified by license start date, but explanation is missing.
	Specify whether the data produced and/or used in the project is useable by third parties, in particular after the end of the project? If the re-use of some data is restricted, explain why		x		license/license_ref	Data usage rights determined by referenced license. Explanation missing.
	Describe data quality assurance processes	x			dataset/data_quality_assurance	Free-form text
	Specify the length of time for which the data will remain re-usable	x			distribution/available_until	The date until when the data should remain available can be specified
Allocation of resources	Estimate the costs for making your data FAIR. Describe how you intend to cover these costs	x			dmp/cost/*	The estimated costs and description can be provided
	Clearly identify responsibilities for data management in your project		x		contributor/role	Roles can be interpreted freely, responsibilities not necessarily clear.
	Describe costs and potential value of long term preservation	x			dmp/cost/*	The estimated costs and description can be provided
Data security	Address data recovery as well as secure storage and transfer of sensitive data	x			dataset/security_and_privacy/*	Free-form text
Ethical aspects	To be covered in the context of the ethics review, ethics section of DoA and ethics deliverables. Include references and related technical aspects if not covered by the former	x			dmp/ethical_issues_exist, dmp/ethical_issues_description, dmp/ethical_issues_report	The Description of Action can be referenced by URL and a description directly in the maDMP provided
Other	Refer to other national/funder/sectorial/departmental procedures for data management that you are using (if any)			x	–	

Table E.1: Assessment to which extent the maDMP can answer the questions of the H2020 DMP template [Eur18].

FWF DMP template questions		Question answerable			Corresponding maDMP fields	Comment
		completely	partially	not		
0 Data Officier	Who is responsible for the data management and the DMP of the project (name/email address)?	x			dmp/contact/*, dmp/contributor/*	Contact person and contributors can be specified with name and email address.Also ORCID and other identifiers, e.g. TISS ID.
I Data Characteristics	What kinds of data/source code will be generated or reused (type, format, volume)?	x			dataset/type, distribution/format, distribution/byte_size	Data type, format and size in bytes can be specified.
	How will the research data be generated and which methods will be used?			x	–	
	How will you structure the data and handle versioning?		x		host/support_versioning	Only describes the capability of the host to support versioning, but not how versioning will be applied to the data. Structuring of data is missing.
	Who is the target audience?			x	–	
II Documentation and Metadata	What metadata standards (if any) will be in use and why? (see Digital Curation Centre)	x			dataset/metadata/*	Metadata standards can be referenced by an URL + free-form text description possible
	What information is needed for the data to be findable, accessible, interoperable and re-usable (FAIR) in the future?		x		dataset/dataset_id, distribution/data_access, distribution/access_url, dataset/metadata/*, distribution/license/*	Information on FAIR, e.g. findability by DOI, metadata, access and license are distributedamong several fields, but not expressive enough to fully answer the question.
	Is the data machine-readable?			x	–	
	How are you planning to document this information?			x	–	
III Data Availability and Storage	What quality assurance processes will you adopt?	x			dataset/data_quality_assurance	Free-form text
	How will the consistency and quality of data collection be controlled and documented? (This may include processes such as repeat samples or measurements, standardised data capture, peer review of data or representation with controlled vocabularies.)	x			dataset/data_quality_assurance	Free-form text
	How and when will the data be shared and made accessible?	x			distribution/*, distribution/host, license/start_date	Location and data host can be specified. License start date describes when data willbe accessible (can specify embargo)
	What repository will you be using?	x			host/url	Repository can be referenced, e.g. re3data repository ID or URL
	What persistent identifier will be used?	x			dataset/dataset_id, host/pid_system	PID for datasets, PID system the host supports
	What data are to be preserved for the long-term, and what data will not be stored?	x			dataset/preservation_statement	Free-form text
	How and where will the data be stored and backed up during the research?	x			host/url, host/storage_type, host/availability, host/backup_type, host/backup_frequency	Describes data location and host with quality of serivce, such as availability, type and frequency of backups
	How and where will the data be stored after the project ends?	x			distribution/access_url, distribution/host/*	Location and data host can be specified
	For how long will the data be stored?	x			distribution/available_until	The date until when the data should remain available can be specified
	Are there any costs that need to be covered for storage?	x			dmp/cost/*	The estimated costs and description can be provided
	At what point during or after the project will the data be stored?		x		license/start_date	License start data can be used to specify when data will be shared. Must not bethe same date as when the data is deposited.
	Are there any technical barriers to making the research data fully or partially accessible?		x		distribution/data_access, dataset/technical_resource	Technical resources for accessing and processing the data can be described which implicitly can impose barriers.
IV Legal andEthical Aspects	Are there any legal barriers to making the research data fully or partially accessible?			x	–	
	Who owns the data?		x		license/license_ref	Data usage rights determined by referenced license. Data ownership could be specified in license.
	What licence for reuse are you planning to attach to the data?	x			license/license_ref	Licenses can be referenced by URI.
	Are there any restrictions on the re-use of the data? If so, why?		x		license/license_ref	Data usage rights determined by referenced license. Explanation missing.
	Are there any ethical barriers to making the research data fully or partially accessible?	x			dmp/ethical_issues_exist, dmp/ethical_issues_description, dmp/ethical_issues_report	Document from meeting with ethics committee can be referenced by URL and a description directly in the maDMP provided
	If applicable, how are you planning to deal with sensitive data during and after the project?	x			dataset/security_and_privacy/*	Free-form text

Table E.2: Assessment to which extent the maDMP can answer the questions of the FWF DMP template [Aus19].

List of Figures

List of Tables

List of Listings

Glossary

active data Data that is collected during the active phase of a study and is subject to change. 9, 10, 29, 35, 47, 48

business process Synonymous to workflow. 3

curation Activity of managing and promoting the use of data from their point of creation to ensure that they are fit for contemporary purpose and available for discovery and reuse[5]. 1, 48

machine-actionable Refers to information that is structured in a consistent way so that machines, or computers, can be programmed against the structure[6]. 3, 10, 15, 107

repository Database or a virtual archive established to collect, disseminate and preserve scientific output like scientific articles and datasets (including software)[7]. 2, 33

research data Factual records used as primary sources for scientific research, and that are commonly accepted in the scientific community as necessary to validate research findings [OEC15]. 1, 33, 47, 72, 73, 103, 127

workflow Sequence of steps in data management planning. Can involve various stakeholders and services. 3, 103–105, 107, 131

[5] https://casrai.org/term/curation/
[6] https://ddialliance.org/taxonomy/term/198
[7] https://www.openaire.eu/where-can-i-read-more-about-fp7

Acronyms

ANDS Australian National Data Service. 3

API Application Programming Interface. 19, 28, 36, 63, 68, 70, 72, 77, 78, 81, 86, 104

BPMN Business Process Model and Notation. 25–27, 36, 37, 39, 50, 92, 104–106, 123

CRIS Current Research Information System. 3, 12, 28, 49, 104

CSS Cascading Style Sheets. 70

CTS Core Trust Seal. 34

DCC Digital Curation Center. 4, 11, 17, 33

DDI Data Documentation Initiative. 3, 15

DMP Data Management Plan. 1, 2, 9

DMR Data Management Record. 19

DOI Digital Object Identifier. 7, 13, 28, 34, 36

DROID Digital Record and Object Identification. 73, 75, 125

EA Enterprise Architecture. 5, 41, 42, 44, 47, 54, 62, 63, 68, 88, 104, 127

EC European Commission. 1, 3, 33

EIRS European and International Research Support. 65

ELN Electronic Lab Notebook. 12, 19

EUDAT European Data Infrastructure. 75, 76, 78, 81, 105, 125

FAIR Findable, Accessible, Interoperable, Reusable. 7, 8, 10, 33, 34, 102

FITS File Information Tool Set. 73–75, 77, 78, 81, 105, 125

FORCE11 The Future of Research Communications and e-Scholarship. 3

FWF Austrian Science Fund. 28, 38, 39, 106

GUI Graphical User Interface. 52, 63, 104

HPC High Performance Computing. 12, 65

HTML Hypertext Markup Language. 70

HTTP Hypertext Transfer Protocol. 70, 77, 87

ICT Information and Communications Technology. 3, 24, 29–31, 37, 42, 65, 68, 82, 86–88, 90, 125

IDCC International Digital Curation Conference. 11

JHOVE JSTOR/Harvard Object Validation Environment. 73, 75, 125

JSON JavaScript Object Notation. 17, 70, 84, 87, 88, 129

maDMP machine-actionable Data Management Plan. 3, 11, 23, 25, 26, 34–36, 38, 39, 123

MIME Multipurpose Internet Mail Extensions. 73, 74, 81, 129

Nectar National eResearch Collaboration Tools and Resources. 88

NSF National Science Foundation. 1

OAI-PMH Open Archives Initiative Protocol for Metadata Harvesting. 28, 35

OpenDOAR Directory of Open Access Repositories. 34

ORCID Open Researcher and Contributor ID. 27, 28, 78, 90

OSBAPI Open Service Broker API. 6, 7, 19, 21, 22, 57, 59, 60, 63, 83–88, 90, 105, 124

OWL Web Ontology Language. 17

PID Persistent Identifier. 10, 12, 26, 28, 34, 123

PUID PRONOM Persistent Unique Identifier. 73, 74, 77, 81, 129

QoS Quality of Service. 21, 30, 35, 50, 57

RDA Research Data Alliance. 3, 14, 17, 33, 78–81, 88, 90, 106, 127

RDM Research Data Management. 2, 23, 25, 26, 42, 62, 63, 102, 103, 105–107, 123

re3data Registry of Research Data Repositories. 13, 15, 34, 68, 72, 76–78, 81, 90, 105, 125

REST Representational State Transfer. 68, 70, 77

RUP Rational Unified Process. 42, 45

SAD Software Architecture Document. 42, 45

SWORD Simple Web-service Offering Repository Deposit. 36

TISS TU Wien Information Systems and Services. 67, 68, 70, 72, 78, 81, 105

TOG The Open Group. 42, 43, 45

TOGAF The Open Group Architecture Framework. 43

UC3 University of California Curation Center. 11, 17

UI User Interface. 5, 6, 37, 70, 104

UML Unified Modeling Language. 45, 53

URI Uniform Resource Identifier. 95, 98, 101

URL Uniform Resource Locator. 86, 98

VM Virtual Machine. 66, 67, 85, 105

XML Extensible Markup Language. 73, 75, 125

Bibliography

[Arm13] Armstrong, Chris and Baker, J.D. and Band, Iver and Courtney, Sam and Jonkers, Henk and Muchandi, Veer and Owen, Martin. Using the ArchiMate Language with UML. White Paper W134, The Open Group, September 2013.

[Aus19] Austrian Science Fund. FWF Data Management Plan Template (DMP) - Guide (01/2019). https://www.fwf.ac.at/fileadmin/files/Dokumente/Open_Access/FWF_DMPTemplate_e.pdf, January 2019. online; last access: March 14, 2020.

[AWG] Data Documentation Initiative ADMP Working Group. Active Data Management Plans: A metadata-driven model for Data Management Plans (Draft White Paper). https://ddi-alliance.atlassian.net/wiki/spaces/DDI4/pages/7864356/. online; last access: March 14, 2020.

[Bal12] Alex Ball. *Review of Data Management Lifecycle Models*. University of Bath, UK United Kingdom, February 2012.

[Bal14] Alex Ball. How to License Research Data. DCC How-to Guides, Edinburgh: Digital Curation Center, July 2014.

[BDW19] Christophe Bahim, Makx Dekkers, and Brecht Wyns. Results of an Analysis of Existing FAIR assessment tools. *Research Data Alliance*. https://doi.org/10.15497/RDA00035, 2019.

[Car14] Jake Carlson. The Use of Life Cycle Models in Developing and Supporting Data Services. *Research Data Management. Practical Strategies for Information Professionals*, pages 63–86, 2014.

[Com16] European Commission. H2020 Programme Guidelines on FAIR Data Management in Horizon 2020. Technical Report Version 3.0, European Commission, Directorate-General for Research & Innovation, July 2016.

[Dat19] DataCite Metadata Working Group. DataCite Metadata Schema Documentation for the Publication and Citation of Research Data. Technical Report Version 4.3, DataCite e.V., 2019.

[DCC] DCC. Funders' data plan requirements | Digital Curation Centre. http://www.dcc.ac.uk/resources/data-management-plans/funders-requirements. online; last access: March 14, 2020.

[DCC13] DCC. Checklist for a Data Management Plan. Technical Report v.4.0, Edinburgh: Digital Curation Center, 2013.

[DCC14] DCC. Five steps to decide what data to keep: a checklist for appraising research data. Technical Report v.1, Edinburgh: Digital Curation Center, 2014.

[ECBW14] Veerle Eynden, Louise Corti, Libby Bishop, and Matthew Woollard. *Managing and Sharing Research Data: A Guide to Good Practice*. SAGE Publications Ltd, January 2014.

[Eur18] European Commission. H2020 templates: Data management plan. Technical Report v2.0, European Commission, Directorate-General for Research & Innovation, February 2018.

[FBR+16] Markus Freudenberg, Martin Brümmer, Jessika Rücknagel, Robert Ulrich, Thomas Eckart, Dimitris Kontokostas, and Sebastian Hellmann. The Metadata Ecosystem of DataID. In Emmanouel Garoufallou, Imma Subirats Coll, Armando Stellato, and Jane Greenberg, editors, *Metadata and Semantics Research*, pages 317–332, Cham, 2016. Springer International Publishing.

[Fou] Queensland Cyber Infrastructure Foundation. ReDBox Data Life Cycle. https://www.redboxresearchdata.com.au/rbdlc/. online; last access: March 14, 2020.

[Fre16] Markus Freudenberg. Data Management Plan extension of DataID. https://wiki.dbpedia.org/use-cases/data-management-plan-extension-dataid, 2016. online; last access: March 14, 2020.

[Gia10] Ronald E. Giachetti. *Design of Enterprise Systems: Theory, Architecture, and Methods*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2010.

[GLJ+18] Marjan Grootveld, Ellen Leenarts, Sarah Jones, Emilie Hermans, and Eliane Fankhauser. OpenAIRE and FAIR Data Expert Group survey about Horizon 2020 template for Data Management Plans. https://doi.org/10.5281/zenodo.1120245, January 2018.

[Gre13] Arofan Gregory. Data Management Planning and the Data Documentation Initiative (DDI). Technical Report Version 1.0, Data Documentation Initiative, 2013.

[Gro17] The Open Group. *ArchiMate 3.0.1 Specification*. Van Haren Publishing, 1st edition, 2017.

[Gro18] The Open Group. *The TOGAF Standard, Version 9.2*. Van Haren Publishing, 11th edition, 2018.

[HBJ19] Rosie Higman, Daniel Bangert, and Sarah Jones. Three camps, one destination: the intersections of research data management, FAIR and Open. *Insights*, 32(1):18, May 2019.

[Hig08] Sarah Higgins. The DCC Curation Lifecycle Model. *International Journal of Digital Curation (IJDC)*, 3(1):134–140, August 2008.

[HJC+18] Simon Hodson, Sarah Jones, Sandra Collins, Françoise Genova, Natalie Harrower, Leif Laaksonen, Daniel Mietchen, Rūta Petrauskaité, and Peter Wittenburg. Turning FAIR into reality: Final report and action plan from the European Commission expert group on FAIR data. `https://doi.org/10.2777/1524`, November 2018.

[HKSP19] Rob Hooft, Mateusz Kuzak, Marek Suchánek, and Robert Pergl. "Data Stewardship Wizard": A Tool Bringing Together Researchers, Data Stewards, and Data Experts around Data Management Planning. *Data Science Journal*, 18(1):59, 2019.

[HT20] Tony Hey and Anne Trefethen. The Fourth Paradigm 10 Years On. *Informatik Spektrum*, 42(6):441–447, January 2020.

[HTT09] Tony Hey, Stewart Tansley, and Kristin Tolle. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, October 2009.

[JG17] Sarah Jones and Marjan Grootveld. How FAIR are your data? `https://doi.org/10.5281/zenodo.1065991`, November 2017.

[Jisa] Jisc. Research Data Management Toolkit | RDM Checklist. `https://rdmtoolkit.jisc.ac.uk/plan-and-design/rdm-checklist/`. online; last access: March 14, 2020.

[Jisb] Jisc. Research Data Management Toolkit | Research data lifecycle. `https://rdmtoolkit.jisc.ac.uk/research-data-lifecycle/`. online; last access: March 14, 2020.

[Jon11] Sarah Jones. How to Develop a Data Management and Sharing Plan. DCC How-to Guides, Edinburgh: Digital Curation Center, 2011.

[JPH+20] Sarah Jones, Robert Pergl, Rob Hooft, Tomasz Miksa, Robert Samors, Judit Ungvari, Rowena I. Davis, and Tina Lee. Data Management Planning: How Requirements and Solutions are Beginning to Converge. *Data Intelligence*, 2(1-2):208–219, 2020.

[Kru95] P. B. Kruchten. The 4+1 View Model of architecture. *IEEE Software*, 12(6):42–50, Nov 1995.

[Lib] UCF Libraries. Scholarly Communication | Overview: Research Lifecycle. https://library.ucf.edu/about/departments/scholarly-communication/overview-research-lifecycle/. online; last access: March 14, 2020.

[MA17] Tomasz Miksa and Kevin Ashley. Workshop report - Research Data Lifecycle and Machine-actionable Data Management Plans. https://doi.org/10.5281/zenodo.1067753, November 2017.

[MCB18] Tomasz Miksa, João Cardoso, and José Luis Borbinha. Framing the scope of the common data model for machine-actionable Data Management Plans. In *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, pages 2733–2742, 2018.

[Mic15] William K. Michener. Ten Simple Rules for Creating a Good Data Management Plan. *PLOS Computational Biology*, 11(10):e1004525, October 2015.

[MNWR18] Tomasz Miksa, Peter Neish, Paul Walk, and Andreas Rauber. Defining requirements for machine-actionable data management plans. In Nance McGovern and Ann Whiteside, editors, *Proceedings of the 15th International Conference on Digital Preservation, iPRES 2018, Boston, MA, USA, September 24-28, 2018*, 2018.

[Mon18] Barend Mons. *Data Stewardship for Open Science: Implementing FAIR Principles*. CRC Press, 2018.

[MR18] Matt McNeeney and Jared Ruckle. An Inside Look at the Open Service Broker API: Easily Deliver Services to Cloud Foundry and Kubernetes. White paper, Pivotal Software, Inc., 2018.

[MRGB17] Tomasz Miksa, Andreas Rauber, Raman Ganguly, and Paolo Budroni. Information Integration for Machine Actionable Data Management Plans. *International Journal of Digital Curation (IJDC)*, 12(1):22–35, September 2017.

[MSMJ19] Tomasz Miksa, Stephanie Simms, Daniel Mietchen, and Sarah Jones. Ten principles for machine-actionable data management plans. *PLOS Computational Biology*, 15(3):e1006750, March 2019.

[MT18] Cox Andrew Martin and Tam Winnie Wan Ting. A critical analysis of lifecycle models of the research process and research data management. *Aslib Journal of Information Management*, 70(2):142–157, March 2018.

[MWN19] Tomasz Miksa, Paul Walk, and Peter Neish. RDA DMP Common Standard for Machine-actionable Data Management Plans. https://doi.org/10.15497/rda00039, 2019.

[NSF18] NSF Directorate for Engineering. Data Management Plans Guidance for Principal Investigators. https://nsf.gov/eng/general/ENG_DMP_Policy.pdf, November 2018. online; last access: March 14, 2020.

[Obl19] Simon Oblasser. Machine-actionable DMP application (DMap). https://doi.org/10.5281/zenodo.3522247, October 2019. The slides are based on a live demo given at RDA 14th Plenary taking place in Helsinki between 22-25 October 2019.

[Obl20a] Simon Oblasser. oblassers/dmap-mockups: DMap Mockups. https://doi.org/10.5281/zenodo.3630375, January 2020. Version 1.2.

[Obl20b] Simon Oblasser. oblassers/tuw-servicebroker: TU.it Service Broker. https://doi.org/10.5281/zenodo.3630401, January 2020. Version 1.0.

[OEC07] OECD. *OECD Principles and Guidelines for Access to Research Data from Public Funding*. OECD Publishing, April 2007.

[OEC15] OECD. Making Open Science a Reality. *OECD Science, Technology and Industry Policy Papers*, (25), October 2015.

[RDA15] RDA. IG Active Data Management Plans Case Statement. Technical Report v3, Research Data Alliance, 2015.

[RDA17] RDA. WG DMP Common Standards Case Statement. Technical report, Research Data Alliance, 2017.

[RVU+15] Jessika Rücknagel, Paul Vierkant, Robert Ulrich, Gabriele Kloska, Edeltraud Schnepf, David Fichtmüller, Evelyn Reuter, Angelika Semrau, and Maxi Kindling. Metadata Schema for the Description of Research Data Repositories. Technical Report Version 3.0, December 2015.

[Sci18a] Science Europe. Practical Guide to the International Alignment of Research Data Management. Technical Report D/2018/13.324/4, Science Europe, November 2018.

[Sci18b] Science Europe. Science Europe Guidance Document Presenting a Framework for Discipline-specific Research Data Management. Technical Report D/2018/13.324/1, Science Europe, January 2018.

[Ser] UK Data Service. Research data lifecycle. https://www.ukdataservice.ac.uk/manage-data/lifecycle.aspx. online; last access: March 14, 2020.

[SJ17] Stephanie Simms and Sarah Jones. Next-Generation Data Management Plans: Global, Machine-Actionable, FAIR. *International Journal of Digital Curation (IJDC)*, 12(1):36–45, September 2017.

[SJM+18] Stephanie Simms, Sarah Jones, Tomasz Miksa, Daniel Mietchen, Natasha Simons, and Kathryn Unsworth. A Landscape Survey of ActiveDMPs. *International Journal of Digital Curation (IJDC)*, 13(1):204–214, December 2018.

[SJMM17] Stephanie Simms, Sarah Jones, Daniel Mietchen, and Tomasz Miksa. Machine-actionable data management plans (maDMPs). *Research Ideas and Outcomes*, 3:e13086, April 2017.

[SSJR16] Stephanie Simms, Marisa Strong, Sarah Jones, and Marta Ribeiro. The Future of Data Management Planning: Tools, Policies, and Players. *International Journal of Digital Curation (IJDC)*, 11(1):208–217, October 2016.

[SUDB18] Nicholas Smale, Kathryn Unsworth, Gareth Denyer, and Daniel Barr. The History, Advocacy and Efficacy of Data Management Plans. *bioRxiv*, 443499, October 2018.

[TU 18] TU Wien. Policy für Forschungsdatenmanagement an der TU Wien. `https://www.tuwien.at/forschung/fti-support/forschungsdaten/forschungsdatenmanagement/policy/`, 2018. online; last access: March 14, 2020.

[WBZ17] Mary Williams, Jacqueline Bagwell, and Meredith Nahm Zozus. Data management plans: the missing perspective. *Journal of Biomedical Informatics*, 71:130–142, July 2017.

[WDA+16] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan Van Der Lei, Erik Van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3:160018, March 2016.