TU WIEN Informatics

Automating Data Management Plan Evaluation Through the Integration of Machine-Actionable Data Management Plans and FAIR Implementation Profiles

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Bernardo Aceves Partida, BSc.
Matrikelnummer 12143531

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Dr.techn. Mag. Tomasz Miksa

Wien, 5. September 2025

Bernardo Aceves Partida | Tomasz Miksa

TU WIEN Informatics

Automating Data Management Plan Evaluation Through the Integration of Machine-Actionable Data Management Plans and FAIR Implementation Profiles

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Bernardo Aceves Partida, BSc.
Registration Number 12143531

to the Faculty of Informatics

at the TU Wien

Advisor: Dr.techn. Mag. Tomasz Miksa

Vienna, September 5, 2025

Bernardo Aceves Partida

Tomasz Miksa

Erklärung zur Verfassung der Arbeit

Bernardo Aceves Partida, 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.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel" habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 5. September 2025

Bernardo Aceves Partida

Danksagung

Zunächst einmal möchte ich meiner Familie und meinem Partner für ihre bedingungslose Unterstützung während der Herausforderungen, denen ich in diesen Jahren gegenüberstand, und dafür, dass sie immer an mich geglaubt haben, danken.

Besonderer Dank gilt meinem Betreuer Tomasz Miksa für seine kontinuierliche Unterstützung und Anleitung, dafür, dass er mir geholfen hat, meinen Horizont mit neuen Ideen und Perspektiven zu erweitern, und dafür, dass er meine Fragen immer umgehend, mit der bestmöglichen Einstellung und höchster Professionalität beantwortet hat.

Außerdem möchte ich mich bei den Freunden bedanken, die ich während dieses Prozesses kennengelernt habe und die mir sowohl Spaß als auch gegenseitige Motivation im akademischen und beruflichen Bereich geboten haben.

Schließlich möchte ich auch allen meinen Professoren für ihr Engagement und ihren Einsatz während meines Studiums danken, deren Anleitung und Lehre wesentlich zu meinem akademischen und persönlichen Wachstum beigetragen haben.

Acknowledgements

First of all, I want to thank my family and my partner for their unconditional support during the challenges I faced over these years and for always believing in me.

Special thanks to my supervisor, Tomasz Miksa, for his continuous support and guidance, for helping me broaden my horizons with new ideas and perspectives and for always answering my questions promptly, with the best possible attitude and the utmost professionalism.

Furthermore, I would like to thank the friends I met during this process who were there for both fun and mutual motivation in the academic and professional spheres.

Finally, I would also like to thank all my professors for their dedication and commitment during my studies, whose guidance and teaching greatly contributed to my academic and personal growth.

Kurzfassung

Kurzfassung

Diese Masterarbeit stellt ein Framework zur automatisierten Bewertung von maschinenlesbaren Data Management Plans (maDMPs) vor. Grundlage ist die Ausrichtung von maDMPs, die nach dem Research Data Alliance Common Standard für maschinenlesbare Data Management Plans (DCS-Schema) strukturiert sind, an community-definierten Erwartungen, die in FAIR Implementation Profiles (FIPs) erfasst werden. Die Arbeit verwendet zwei zentrale Metriken: Erstens die mapping coverage, die den Anteil der Fragen misst, die zur Erstellung eines FIP genutzt werden und Feldern des DCS-Schemas zugeordnet werden können; zweitens das Konzept der planned FAIR alignment, das bewertet, wie gut die in einem maDMP angegebenen oder geplanten Maßnahmen (z. B. Wahl der Lizenz, Metadatenschema, Identifikatoren) mit den Erwartungen der FIPs übereinstimmen. Im Unterschied zu Werkzeugen zur FAIR-Bewertung auf Datensatzebene konzentriert sich dieser Ansatz auf die *Compliance* innerhalb von RDM-Workflows.

Ein Prototyp zur Compliance-Prüfung implementiert diese Zuordnungen und erzeugt strukturierte Evaluationsergebnisse, die dem OSTrails FAIR Test Result Specification (FTR) entsprechen. Diese Ergebnisse unterstützen semantische Abfragen in RDF und erleichtern automatisierte Vorab-Prüfungen in Aufgaben des Forschungsdatenmanagements (z. B. Vollständigkeitsprüfungen und Richtlinienabgleich mit Community-Profilen). Das Framework unterstützt zudem die Übernahme von FIP-Fragen aus Nanopublications, um neue Zuordnungen in das Tool einzubinden und so die Übertragbarkeit auf verschiedene Fachdomänen zu verbessern. Insgesamt trägt die Arbeit mit einem operativen Werkzeug und einem methodischen Ansatz dazu bei, FAIR-orientierte Planung zu stärken und eine reproduzierbare, maschinenlesbare Bewertung von maDMPs zu ermöglichen.

Abstract

This master's thesis presents a framework for the automated assessment of machine-actionable Data Management Plans (maDMPs). It is based on aligning maDMPs structured via the Research Data Alliance Common Standard for machine-actionable Data Management Plans (DCS) schema, with community-defined expectations, captured in FAIR Implementation Profiles (FIPs). The research employs two central metrics: first, the mapping coverage, measuring the proportion of the questions used to create a FIP that can be matched to fields in the DCS schema; second, the concept of planned FAIR alignment, evaluating how well the actions (or planned actions) specified in a maDMP (e.g., choice of license, metadata schema, identifiers) align with FIP expectations. Unlike dataset-level FAIR scoring tools, the approach focuses on *compliance* within Research Data Management (RDM) workflows.

A prototype compliance checker implements these mappings and produces structured evaluation outputs conforming to the OSTrails FAIR Test Result specification FTR. These outputs support semantic querying RDF and facilitate automated pre-assessment workflows for data stewardship tasks (e.g., completeness checks and policy alignment against community profiles). The framework also supports ingesting FIP questions from Nanopublications to add new mappings to the tool, improving portability across domains. Overall, the thesis contributes an operational tool and a methodological approach that strengthen FAIR-aligned planning and enable reproducible, machine-readable assessment of maDMPs.

Contents

CHAPTER 1

Introduction

1.1 Context

The rapid increase in data production across research disciplines has highlighted the importance of effective data management to facilitate reproducibility, transparency, and reusability of research outcomes [WDA+16]. To address these challenges, the concept of Research Data Management RDM has emerged as a crucial practice within the scientific community, promoting structured processes for data collection, curation, preservation, and sharing [Mon17].

An essential component of RDM is the preparation and maintenance of Data Management Plans (DMPs). Traditional DMPs are static, text-based documents, often treated as administrative obligations rather than useful research tools [SJMM17]. However, with the introduction of machine-actionable Data Management Plans (maDMPs), the research community has begun transitioning toward more dynamic and interactive DMPs that integrate into digital research infrastructures [MWN19].

Machine-actionability refers to the capability of digital entities (e.g., maDMPs) to be interpreted, processed, and acted upon by software, facilitating automation, interoperability, and scalability in data management tasks [MWN19, MWN+21]. The Research Data Alliance (RDA) has significantly contributed by publishing the RDA Common Standard for maDMPs DCS [Res21], providing a structured representation for essential DMP components using common vocabularies and JSON-based schemas [Res21].

Simultaneously, the scientific community has increasingly embraced the FAIR principles—Findability, Accessibility, Interoperability, and Reusability—as a gold standard for data stewardship [WDA+16]. Despite their wide adoption, operationalizing FAIR principles remains challenging, primarily due to their high-level nature. To bridge this gap, FAIR Implementation Profiles (FIPs) were proposed, enabling communities to specify precise, actionable steps for implementing FAIR principles in practice [SM+20].

1.2 Problem Statement and Motivation

Despite these advancements, evaluating the quality of maDMPs and ensuring compliance with community-specific FAIR standards remains largely a manual and time-consuming task [Arn24, TP23]. Most existing methods for DMP evaluation involve subjective, textual assessments by reviewers, leading to inconsistency and inefficiency [SJMM17, MWN19]. Consequently, there is a critical need for automated tools capable of systematically assessing maDMP quality against established standards and guidelines, such as those defined by FAIR Implementation Profiles [SM+20].

Moreover, integrating structured FAIR assessment outputs into existing research infrastructure remains challenging due to a lack of standardized reporting formats. Recent initiatives, such as the Open Science Training and Resources for Advanced Interoperable Learning Systems (OSTrails), aim to address this issue by defining standardized FAIR assessment outputs [GW+25]. Leveraging such standards in the context of maDMP evaluation can greatly enhance interoperability, transparency, efficiency and usability of evaluation outcomes.

1.3 Goals and Scope

The primary goal of this thesis is to design and implement an automated evaluation framework for maDMPs, leveraging FIPs and aligning with the OSTrails FAIR assessment specification [GW+25]. Specifically, this thesis aims to:

- Develop a mapping strategy between the questions to create a FIP [GO nd] and the DCS schema to enable automated compliance checks.
- Implement an automated compliance checker capable of assessing critical quality indicators (e.g., completeness, feasibility, and guideline compliance) [Arn24].
- Export the evaluation outcomes in a structured, semantic, and interoperable format using the OSTrails specification [GW+25].
- Demonstrate practical applicability through a semantic web approach, leveraging RDF representations in TTL format for integration with FAIR-compliant infrastructures.

This thesis explicitly does not address real-time integration with external FAIR assessment services (e.g., F-UJI API integration), leaving such advanced integrations for future research.

1.4 Research Questions

The main objective of this thesis is to assess how FIPs can be leveraged to automate the evaluation of maDMPs using the DCS schema.

This objective is broken down into the following research questions:

Research question 1: To what extent can machine actionable Data Management Plans (maDMPs) and FAIR Implementation Profiles (FIPs) be aligned to enable automated evaluation of Data Management Plans?

- This question investigates the degree to which fields in the DCS schema can be mapped to the specific questions to create a FIP, allowing for automated compliance checks.
- The extent of this alignment is quantified using the metric of mapping coverage; defined as the proportion of FIP questions that have corresponding fields in the DCS schema.
- To further explore the utility of this mapping, the following subquestion is posed:
  - **Research question 1.1**: Which of the DMP evaluation goals can be supported by FIPs, and to what extent can these be automated?
- The goals considered include completeness, feasibility, and guideline compliance; drawn from recent literature on DMP assessment [Arn24, TP23] (see subsection 2.8.1.

Research question 2: To what extent can the intended actions in maDMPs be evaluated against FAIR-enabling expectations defined in community-specific FIPs? Particularly when data has not yet been generated or lacks repository identifiers.

- Early-stage DMPs typically outline planned decisions (e.g., licenses, identifier types, metadata schemas) rather than refer to existing datasets.
- This question investigates whether these planned actions align with the expectations specified in a relevant FIP, based on a rule-based comparison of field values.
- This alignment, referred to as planned FAIR alignment, is treated as a proxy indicator of whether the future data management practices are likely to conform to community expectations.

Addressing these questions will produce a robust automated evaluation framework, which will facilitate practical guidance to researchers and improve data management practices. It is also important to establish that this thesis does not perform FAIR scoring of actual datasets, external tools such as F-UJI are considered in the discussion to illustrate how such evaluations could complement planned-action assessments in future work.

1.5 Methodology

This thesis employs a **Design Science Research** (DSR) methodology, following established guidelines for developing and evaluating artifacts in Information Systems research [HMPR04]. Design science is particularly suitable when creating practical solutions (artifacts) for real-world problems, rather than only describing phenomena [con25].

The research process is structured into iterative stages, aligned with frameworks such as the Three-Cycle View (relevance, design, and rigor) [con25], and the six-step methodology proposed by Peffers et al. [PTG+20]:

1. **Requirements analysis and contextualization**
    - Analyze the structure of the DCS schema, OSTrails FAIR assessment specification, and exemplar FIPs.
    - Identify the key evaluation goals: completeness, feasibility and guideline compliance.
2. **Mapping construction and measurement of alignment coverage**
    - Map FIP questions to DCS schema fields.
    - Compute the mapping coverage, i.e., the proportion of FIP questions covered—quantifying RQ1's *extent* metric.
3. **Prototype design: Compliance checker implementation**
    - Implement a Python-based tool that compares maDMP fields with FIP expectations using rule-based logic.
4. **Evaluating planned FAIR alignment**
    - Assess the degree of planned FAIR alignment, the alignment of intended actions in maDMPs with FIP-defined expectations—addressing RQ2 as a proxy evaluation strategy.
    - Conceptually situate this measurement as a *proxy indicator*—a standard approach in empirical research when direct measurement is not possible [con24].
5. **Structured output and semantic representation**
    - Serialize evaluation outcomes in the OSTrails JSON-LD format for FAIR assessment results.
    - Convert to RDF and demonstrate SPARQL-based querying to enable integration with knowledge graph infrastructures.

6. **Demonstration and artifact evaluation**
   - Apply the compliance checker to real maDMPs to demonstrate feasibility and utility.
   - Manually use F-UJI to assess actual datasets and illustrate how the artifact could complement existing FAIR assessment tools (as future work) [DHF25].

1.6 Thesis Structure

The remainder of this thesis is structured as follows:

- **Chapter 2: Background and Related Work** introduces the foundational concepts that underpin this research, including the FAIR principles, maDMPs, FIPs, and relevant Semantic Web technologies. It also reviews prior approaches to DMP evaluation, FAIR assessment tools, and community-driven initiatives such as GO FAIR and the RDA, situating this thesis within the broader state of the art.
- **Chapter 3: Research Methodology** systematically identifies and justifies the requirements for an automated evaluation framework. This includes functional requirements (e.g., mapping maDMP fields to FIP expectations), non-functional requirements (e.g., interoperability and extensibility), and alignment with established evaluation goals (e.g., completeness, feasibility, and guideline compliance).
- **Chapter 4: Conceptual Design** describes the proposed approach in detail. It elaborates on the mapping strategy that links FIP questions to DCS fields, the design of evaluation rules, and the use of community vocabularies and registries. The framework is also presented as an architectural model that highlights the interplay between inputs, validation rules, and standardized outputs.
- **Chapter 5: Implementation** provides a detailed account of the compliance checker, focusing on the modular structure of the codebase, the integration of external resources, such as SPDX licenses, and the serialization of results using the OSTrails specification. This chapter also discusses the conversion of JSON-LD results into RDF for semantic querying.
- **Chapter 6: Evaluation and Results** demonstrates and analyzes the outcomes of applying the framework to representative maDMP examples and community-specific FIPs. The analysis addresses **RQ1** by measuring the coverage of FIP questions that can be mapped to DCS fields, and **RQ2** by assessing the alignment of intended actions in the maDMP with community FAIR-enabling expectations. As an illustration of possible extensions, an exploratory comparison is made with external FAIR evaluation tools (e.g., F-UJI), highlighting how dataset-level FAIRness assessments could complement planning-time compliance checks. Results are presented in tables and figures and discussed against evaluation goals and prior work.

- **Chapter 7: Conclusion and Future Work** summarizes the findings of the research, highlighting the main contributions to both theory and practice. It reflects on the limitations of the current implementation, outlines recommendations for stakeholders such as researchers and funders, and suggests avenues for future work, including integration with external FAIR services and broader adoption across domains.

CHAPTER 2

Background and Related Work

2.1 Research Data Management and Data Management Plans

Research Data Management (RDM) comprises the policies, processes, and technologies that govern how research data are created, documented, stored, shared, and preserved across the research lifecycle. A Data Management Plan (DMP) formalizes these decisions, typically covering data types and volumes, formats, documentation and metadata practices, storage and backup, access control, licensing, and long-term preservation. Funders and institutions increasingly require DMPs to promote transparency, reproducibility, and reuse [MWN19].

Historically, DMPs were static, free-text documents, which limited their reusability and integration with other infrastructures. The community response has been the shift toward machine-actionable DMPs (maDMPs), which represent DMP content in structured formats so software can interpret, validate, and exchange information automatically [MWN19]. To underpin interoperability, the RDA DMP Common Standard (DCS) defines a JSON application profile for expressing core DMP entities (projects, datasets, storage, licenses, repositories, etc.) with controlled structure and semantics (see Figure 2.1) [Res21]. This standard enables different tools (e.g., DMPTool, Data Stewardship Wizard) to produce and consume maDMPs consistently.

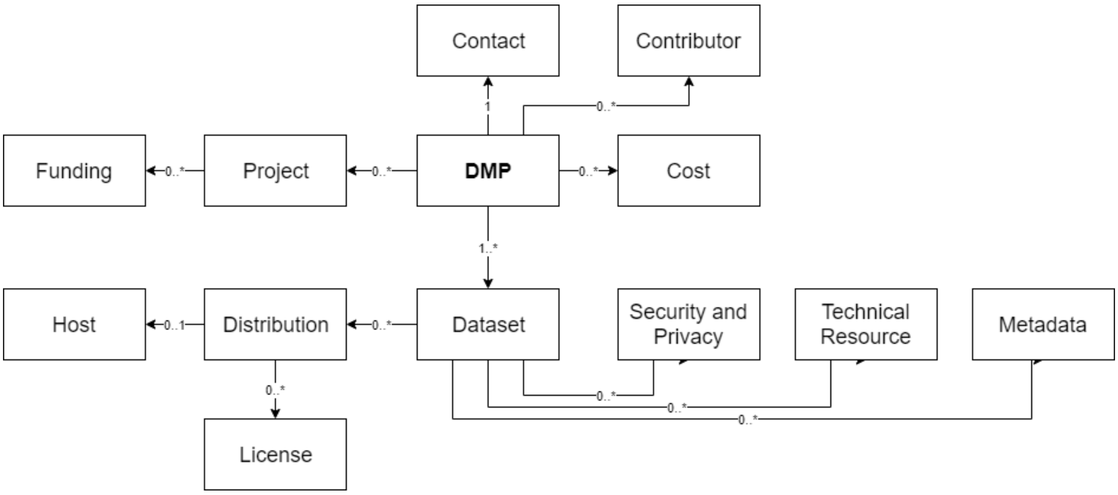

Figure 2.1: RDA DMP Common Standard diagram for machine-actionable Data Management Plans. [Res21]

2.1.1 Ten Principles for maDMPs

The ten principles summarize expectations for machine-actionability in DMPs and inform current community practice [MWN19]. They focus on integrating DMPs into research workflows. They emphasize interoperable data models, automation, and availability for humans and machines. The goal is to reduce administrative burden (from managing free-form text documents) and facilitate automated data management processes in the research ecosystem. The ten principles are shown in Figure 2.2. In summary, they:

- **Integrate** DMPs with workflows to enable automation,
- **Use** PIDs, controlled vocabularies, and a common maDMP data model,
- **Publish** DMPs for human/machine use; monitor and version them,
- **Support** evaluation and policy feedback loops.

2.2 FAIR Principles and FAIR Data

The FAIR Guiding Principles: Findable, Accesible, Interoperable, and Reusable; proposed by Wilkinson *et al.* [WDA+16] have become a widely endorsed compass for data stewardship. Their central aim is to ensure that both humans and machines can discover, access, and make effective use of data with minimal friction. The principles are deliberately high-level and technology-agnostic: they define *what* constitutes good practice (e.g., persistent identifiers, rich metadata, community standards) without prescribing *which* technologies to use. This fosters domain-specific innovation, but also results in

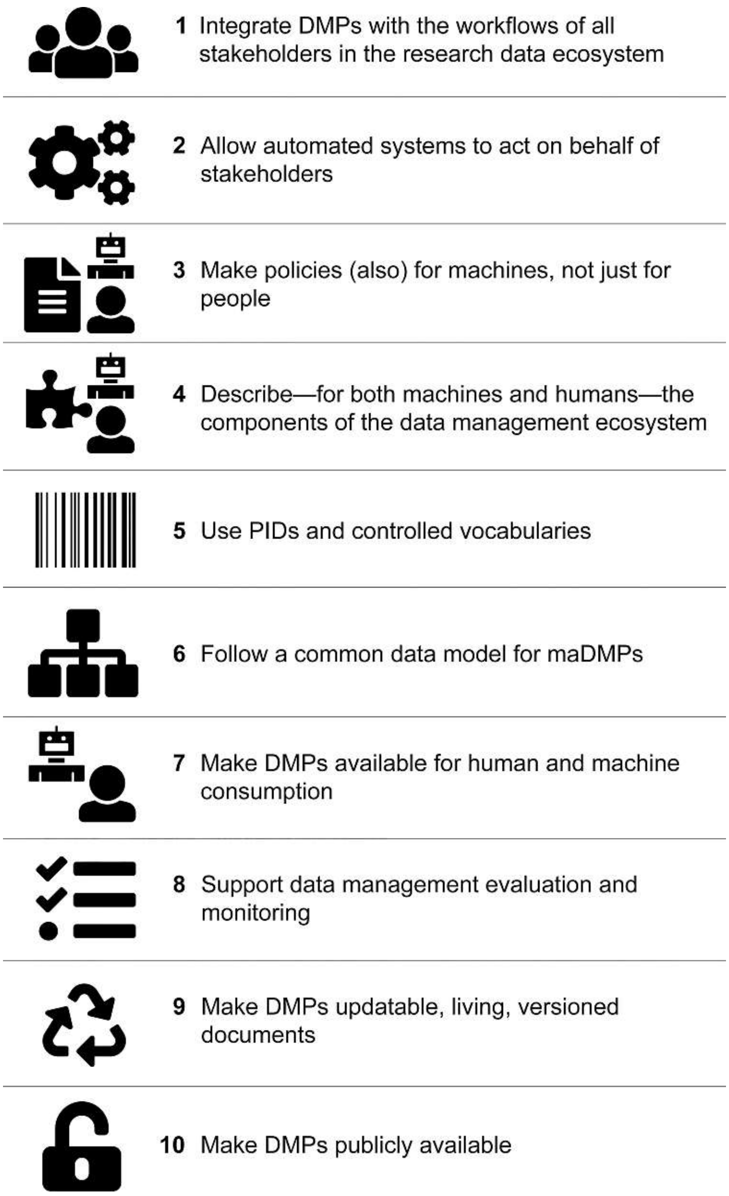

Figure 2.2: Ten principles for maDMPs [MWN19].

heterogeneous implementations across communities. They are divided into four main categories, each with specific sub-principles:

Findable (F)

For data to be reusable, they must first be easy to locate by both humans and computaters. The Findability principle defines the use of persistent identifiers and rich metadata to enable reliable discovery across systems and search engines.

- **F1.** (Meta)data are assigned a globally unique and persistent identifier.
- **F2.** Data are described with rich metadata.
- **F3.** Metadata clearly and explicitly include the identifier of the data they describe.
- **F4.** (Meta)data are registered or indexed in a searchable resource.

Accessible (A)

Once data are findable, they should be retrievable in a standardized way. Accessibility focuses on the use of open, universally implementable protocols and ensures that metadata remain available even if the data are not.

- **A1.** (Meta)data are retrievable by their identifier using a standardized communications protocol.
  - **A1.1.** The protocol is open, free, and universally implementable.
  - **A1.2.** The protocol allows for authentication and authorization when necessary.
- **A2.** Metadata are accessible, even when the data are no longer available.

Interoperable (I)

To support data exchange and integration, interoperability emphasizes the use of standardized formats, vocabularies, and links between datasets. This ensures that data from different sources can be combined and understood consistently.

- **I1.** (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- **I2.** (Meta)data use vocabularies that follow FAIR principles.
- **I3.** (Meta)data include qualified references to other (meta)data.

Reusable (R)

The ultimate goal of FAIR is to maximize reuse of data. Reusability requires rich contextual information, clear licensing, provenance, and adherence to community standards to ensure that data can be trusted and appropriately cited in future research.

- **R1.** (Meta)data are richly described with a plurality of accurate and relevant attributes.
  - **R1.1.** (Meta)data are released with a clear and accessible data usage license.
  - **R1.2.** (Meta)data are associated with detailed provenance.
  - **R1.3.** (Meta)data meet domain-relevant community standards.

Together, these sub-principles form a layered framework: starting with unique identifiers **(F1)**, ensuring retrievability through open protocols **(A1)**, enabling cross-system understanding via shared vocabularies **(I1-I3)**, and culminating in trust and reuse through licensing and provenance **(R1-R3)**. Importantly, the principles emphasize *machine-actionability*, i.e., the capacity of computational systems to autonomously discover and process data [WDA+16, Mon17]. Communities and infrastructures operationalize these principles by defining specific standards and services—choices which are then captured in FIPs.

2.3 FAIR Implementation Profiles (FIPs)

Because FAIR is principle-based, communities need a way to *declare* the concrete technologies they use to fulfill each principle—identifier services, metadata schemas, vocabularies, repositories, access protocols, license frameworks, and so forth. FAIR Implementation Profiles (FIPs) capture these community choices in a structured, reusable way [SM+20]. Conceptually, a FIP is a set of FAIR-enabling resource FER selected by a community of practice to implement specific FAIR sub-principles (e.g., F1 via DOI from DataCite; I1 via RDF/JSON-LD using DCAT; R1.1 via Creative Commons licenses). Tables 2.1 and 2.3 show the GO FAIR template for the creation of a FIP[GO nd].

A FIP ontology (FIP ontology) [MSK25] formalizes the entities (FIP, declaration, FAIR sub-principle, FAIR-enabling resource, resource type) and relations (e.g., *declares-use-of*, *addresses-principle*) so that profiles are interoperable and computable [SM+20]. The Data Stewardship Wizard (DSW) supports communities in producing FIPs through a guided questionnaire using their FIP wizard tool [GO 24]; the resulting declarations can be published as *nanopublications* to facilitate provenance, citability, and integration in linked data ecosystems [SM+20]. Nanopublications represent small, citable RDF graphs (assertion, provenance, publication info), which suits granular FAIR declarations.

Table 2.1: FIP creation template (part 1)[GO nd]

Community description	
Name of Community	e.g., ENVRI
Description of Community	
Supporting Links	
Research Domain	e.g., Environmental Sciences
Data Steward	e.g., ORCID #
Date of FIP creation	

Table 2.2: FIP creation template (part 2a)[GO nd]

FAIR Principle	Question	FAIR enabling resource types	Your answers
F1	What globally unique, persistent, resolvable identifier service do you use for metadata records?	Identifier service	e.g., PURL, DOI
F1	What globally unique, persistent, resolvable identifier service do you use for datasets?	Identifier service	
F2	Which metadata schemas do you use for findability?	Metadata schema	
F3	What is the schema that links the persistent identifiers of your data to the metadata description?	Metadata-Data linking schema	
F4	In which registry are your metadata records indexed?	Registry	
F4	In which registry are your datasets indexed?	Registry	
A1.1	Which standardized communication protocol do you use for metadata records?	Communication protocol	
A1.1	Which standardized communication protocol do you use for datasets?	Communication protocol	
A1.2	Which authentication & authorisation service do you use for metadata records?	Authentication & authorisation service	
A1.2	Which authentication & authorisation service do you use for datasets?	Authentication & authorisation service	
A2	Which metadata preservation policy do you use?	Metadata preservation policy	

Table 2.3: FIP creation template (part 2b)[GO nd]

FAIR Principle	Question	FAIR enabling resource types	Your answers
I1	Which knowledge representation languages (allowing machine interoperation) do you use for metadata records?	Knowledge representation language	
I1	Which knowledge representation languages (allowing machine interoperation) do you use for datasets?	Knowledge representation language	
I2	Which structured vocabularies do you use to annotate your metadata records?	Structured vocabulary	
I2	Which structured vocabularies do you use to annotate your datasets?	Structured vocabulary	
I3	Which models, schema(s) do you use for your metadata records?	Semantic model	
I3	Which models, schema(s) do you use for your datasets?	Semantic model	
R1.1	Which usage license do you use for your metadata records?	Data usage license	
R1.1	Which usage license do you use for your datasets?	Data usage license	
R1.2	Which metadata schemas do you use for describing the provenance of your metadata records?	Provenance model	
R1.2	Which metadata schemas do you use for describing the provenance of your datasets?	Provenance model	

2.4 Linking FIPs and maDMPs

The convergence of FIPs and maDMPs appears in two complementary directions in current literature:

- **FIP-informed DMP authoring.** Communities can use FIPs to guide DMP creation by pre-filling or constraining fields (e.g., licenses, identifier systems, metadata schemas) so that plans reflect community best practices from the outset. The FIP2DMP approach operationalizes this within the Data Stewardship Wizard by mapping FIP declarations to template fields [HMG+23].

- **Evaluating DMPs with structured checks.** Parallel work outlines automated assessment of maDMPs—for schema conformance, completeness, consistency, and policy alignment—leveraging their structured nature [TP23]. While the literature does not yet establish a standardized method to evaluate maDMPs directly against FIP expectations, the ingredients are present: computable FIPs [SM+20] and interoperable DCS-based maDMPs [Res21]. This suggests a promising avenue for rule-based alignment checks in the planning stage.

This thesis targets the second direction: using FIPs as a benchmark for *automated evaluation* of maDMPs, reporting where plans align or deviate and surfacing actionable feedback.

2.5 OSTrails and interoperability for FAIR assessments

Automated checks require interoperable inputs and outputs. The OSTrails initiative proposes an interoperability reference architecture connecting DMP platforms, SKGs, and FAIR assessment services. Within OSTrails, the FAIR Interoperability Framework (FAIR-IF) defines the FAIR Test Results (FTR) model and APIs, enabling FAIR assessment tools to emit results in a shared JSON-LD structure that extends and reuses DCAT, DQV, and PROV concepts [OST25b, GO25]. In parallel, the DMP Interoperability Framework DMP-IF builds on the DCS to expose maDMPs via consistent APIs, supporting real-time updates (e.g., inserting a DOI once a dataset is registered) and feeding evaluations back into the plan [OST25a, OST25c].

2.5.1 Core entities in the FAIR Test Results (FTR) vocabulary.

OSTrails' FTR vocabulary provides a common language to describe assessment main concepts:

- **Metric**: A narrative, domain-agnostic description that a *Test* must wholly implement.

- **Test**: A service (API + code) that implements a Metric and returns a standardized result.
- **TestResult**: The output of running a Test over a resource; it extends *prov:Entity* and points to the corresponding Test via *ftr:outputFromTest*. Test results include status and provenance metadata.
- **TestResultSet**: A collection of *TestResult*s with common metadata (e.g., all results produced by one API call).
- **TestExecutionActivity**: The action (a *prov:Activity*) that produced a *TestResult* or *TestResultSet*.
- **Benchmark**: A community-defined grouping of Metrics that narrate how a domain interprets FAIR for assessment purposes.
- **Algorithm**: Code that contextualizes all results for a given Benchmark into an aggregate.

Versioning note. Unless stated otherwise, this thesis adopts the FTR vocabulary **version 1.1.0**; each of the above concepts may include additional optional properties/components beyond those cited here. The detailed *minimum* set enforced by our design is specified in Chapter 3.

In practice, these frameworks support a DMP evaluation service that parses a maDMP, runs internal policy/completeness checks, optionally invokes external FAIR tools (e.g., F-UJI) for datasets referenced in the plan, and publishes results as OSTrails-compliant FTR reports to enable aggregation, visualization, and cross-tool comparison [OST25b, OST25c]. The OSTrails reference architecture explicitly positions these components across DMP platforms, SKGs, and FAIR services so they can exchange inputs/outputs predictably.

2.6 Semantic Web Foundations for maDMP/FIP Alignment

Semantic Web technologies provide a well-established substrate for representing maDMPs, FIPs, and assessment outcomes as interlinked, machine-readable knowledge:

- **RDF and JSON-LD.** RDF is the standard graph data model for expressing resources and relationships on the Web [CWL14]. JSON-LD enables JSON documents to carry RDF semantics via a context, facilitating exchange between Web APIs and knowledge graphs [SKL20]. This combination is widely used to align system-specific JSON (e.g., DCS serializations) with community vocabularies and repository schemas.

- **SPARQL querying.** SPARQL is the W3C query language for RDF graphs, supporting expressive graph pattern matching and aggregation [HS13]. Once maDMP content, FIP declarations, and assessment outcomes are available as RDF, SPARQL can retrieve alignment status, detect missing fields, and join with external SKGs (e.g., to resolve a DOI to provenance records).

- **Nanopublications.** In current FIP tooling, FIP declarations can be published as nanopublications (via the FIP Wizard), and each nanopublication carries an assertion together with its provenance and publication info. This supports per-decision provenance and citability and can be leveraged by evaluation workflows to reference the exact community recommendation being checked [SM+20, SMK+22, GO 24, FAI24].

- **DMP Common Standard Ontology** DCSO Beyond the JSON profile of the DCS, DCSO provides a formal ontology for maDMPs, enabling RDF serializations and shape validation.[CCE+22]

Together, these technologies support transparent and explainable alignment checks: assessment outputs can point to specific FIP declarations and to precise DCS paths inspected, while results are serialized as linkable resources suitable for integration and reuse.

2.7 FAIR Assessment Tools and Quality Indicators

A range of FAIR assessment tools and maturity models translate FAIR principles into measurable indicators. Among automated tools, F-UJI executes machine tests against dataset landing pages and repository APIs to check for PIDs, rich metadata, licenses, protocol use, and standard vocabularies, returning structured results [DH21]. Other tools and frameworks (e.g., RDA FAIR Data Maturity Model) provide indicator sets that can be aligned with community practices and repository capabilities. While these tools typically evaluate *existing* datasets post hoc, FIP-based planning-time checks (as in this thesis) assess whether intended choices in a maDMP *align* with FAIR-enabling expectations.

To operationalize FAIR, multiple initiatives emerged. GO FAIR promotes community-driven, bottom-up FAIR adoption through Implementation Networks and shared practices [SM+20]. FAIR assessment efforts (e.g., maturity models and automated evaluators) [WDS+19] translate the high-level principles into practical indicators and tests, enabling objective checks of FAIR-supporting behaviors. Together, these efforts build an ecosystem in which DMP planning, community standards, and automated assessment can reinforce each other.

2.8 Automated DMP Evaluation: Prior Work

Miksa *et al.* surveyed case studies and proposed a roadmap toward automated assessment of maDMPs—including validation against schemas, linking to repositories and SKGs, and leveraging FAIR evaluators where identifiers exist [TP23]. They argue that structured maDMPs are key to reducing manual reviewer burden and enabling automated consistency, completeness, and policy checks. Arnhold's thesis implemented a prototype DMP evaluation service to generate automated indicators for completeness, feasibility, and guideline compliance, and studied the partial automation of funder rubrics [Arn24]. Together, these efforts motivate an architectural pattern: DCS-based parsing and validation, rule-based checks against community or funder expectations, optional calls to external FAIR tools, and explainable, structured outputs.

Miksa *et al.* also examine usage scenarios and requirements for automated maDMP evaluation. Figure 2.3 illustrates the current situation (AS-IS) alongside two envisioned approaches for automating the review process (TO-BE-1 and TO-BE-2). In the AS-IS setting, even when a DMP exists as a machine-actionable version, it is typically transformed into a text-based document that follows the funder's template and is then sent to the reviewer. As a result, the advantages of maDMP remain unused in the review workflow, creating additional manual effort and producing feedback that largely depends on the reviewer's individual expertise.

In TO-BE-1, researchers continue to follow their existing workflows to produce a maDMP, which is then submitted to the funder. The funder employs dedicated evaluation software that generates both structured and human-readable outputs, providing metrics that support reviewers and lessen the need for extensive manual background checks. If a funding body chooses not to deploy such software, researchers can still gain value from pre-submission checks, as illustrated in TO-BE-2, where the DMP undergoes automated evaluation before being delivered. However, in general there is still no standardized architecture that systematically exploits maDMPs to automate evaluation and assist reviewers.

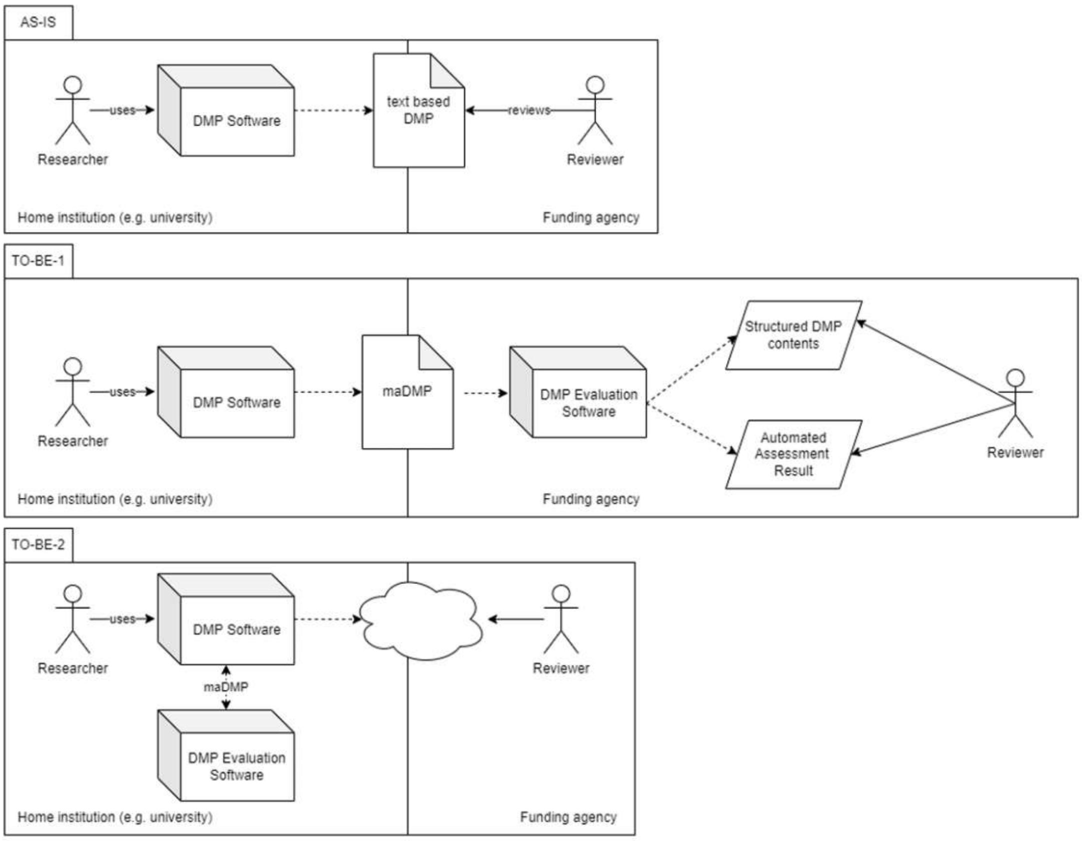

Figure 2.3: Scenarios depicting current situation (AS-IS) and two possible scenarios for automation of maDMP evaluation (TO-BE). [TP23]

2.8.1 Evaluation goals

Following Miksa et al and . [TP23], DMP review goals can be summarized as:

- **G1 completeness**: Whether the maDMP contains all required or conditionally required elements and values needed to support assessment. This covers structural presence per DCS (e.g., required paths exist), and referential presence (e.g., if a repository is named, its identifier and access conditions are provided).
- **G2 feasibility**: Whether the planned actions in the maDMP can realistically be carried out given the available infrastructure and policy constraints. We assess three components: accuracy (choices match allowed catalogs—PID schemes, repositories, licenses), availability (referenced services/resources exist and are accessible), and consistency (no cross-field contradictions).
- **G3 Quality of actions**: Relevance/effectiveness of the planned actions, often linked to FAIR-enabling practices.

- **G4 Guideline compliance**: Replaces "non-ambiguity" from Miksa et al. Following Arnhold [Arn24], we define G4 as alignment with domain specific guidelines, because the original "non-ambiguity" goal targets narrative text and free-text justifications that are not machine-actionable. This replacing keeps the evaluation automatable on structured fields while leaving linguistic clarity outside scope.

2.9 Synthesis and Open Gaps

The field has converged on several pillars:

- **Structured planning input:** maDMPs expressed with DCS enable reliable parsing, validation, and integration [Res21].
- **Community expectations:** FIPs encode FAIR-enabling resource choices per principle in a computable form [SM+20].
- **Assessment interoperability:** OSTrails FAIR-IF and the FAIR Test Results model standardize outputs for cross-tool comparison and feedback loops to DMPs [GW+25].
- **Semantic integration:** RDF/JSON-LD and SPARQL enable linking maDMPs, FIPs, and results, with nanopublications preserving provenance [SM+20].

Open gaps remain: Robust, reusable mappings between FIPs and DCS fields across domains; Systematic, explainable planning-time evaluations that provide actionable guidance before datasets exist; Harmonized result formats and dashboards for stakeholders. This thesis addresses these gaps by operationalizing FIP-to-DCS mappings, implementing rule-based checks over maDMPs, and emitting OSTrails-compliant, semantics-ready results to support integrated, automated review.

While existing tools provide valuable insights, it's important to note that they primarily focus on evaluating existing datasets. Also, Integrating FAIR assessments into the planning stages of research projects remains an area for further development.

CHAPTER 3

Conceptual Design

The goal of this chapter is to present the conceptual design and overall system architecture of the automated maDMP evaluation tool. We describe the main components, their interactions, and the data flow from maDMP input through FIP-constrained tests to OSTrails-compliant outputs. The design is driven by the stakeholder requirements identified in Section 3.1 and operationalizes, in an automated and modular way, the evaluation goals of completeness, feasibility, and guideline compliance. Finally, we define the external interfaces (CLI and API), the catalogs of constraints (e.g., identifier schemes, licenses, metadata standards), and the metric/test/benchmark model that together enable reproducible assessments with evidence to support them.

3.1 Requirements Engineering

In this section, we collect and structure the requirements for automating the evaluation of maDMPs against community-specific FAIR Implementation Profiles (FIPs). The main outputs are OSTrails-compliant JSON-LD results, optionally converted to TTL for semantic querying. We start with who interacts with DMPs, what they need, and where automation helps. From there we derive use cases and requirements; subsequent sections of this chapter then introduce the conceptual mapping to the DCS schema and the evaluation protocol.

3.1.1 Stakeholders and Roles

We adopt the stakeholder landscape from the Ten Principles for maDMP [MWN19] and related work [TP23, Arn24]. Actors include researchers, stewards, funders, repositories, publishers, and ethics/legal experts, as ilustrated in Figure 3.1 We group them into three **roles**:

- **DMP Maintainer:** Researchers and stewards who author or update an DMP. They benefit from automated completeness checks and recommendations.
- **Reviewer:** Funders, ethics boards, legal experts, institutional administrators, or publishers who require consistent, inspectable guideline-compliance metrics to support decisions.
- **Review System Facilitator:** Experts (research support staff, repository operators, infrastructure providers) who configure FIPs, mappings, and benchmarks.

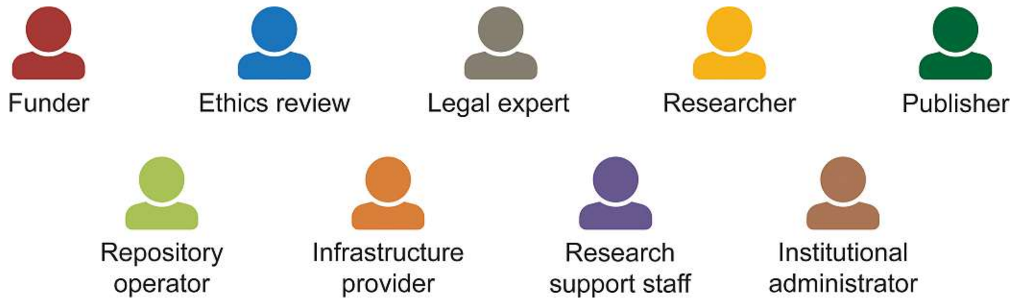

Figure 3.1: Stakeholders involved in creation, review, and governance of DMPs (maDMPs target audience) [MWN19]

3.1.2 Scope

We focus on methods dealing with machine-actionable information (and explicitly excluding language checks of non-structured text), we concentrate on goals G1, G2 and G4 (see subsection 2.8.1).

- **Coverage and compliance checks.** The method evaluates how well FIPs questions can be represented in maDMPs (*mapping coverage*) and whether the resulting maDMP content complies with the constraints and expectations defined in the selected FIP (*guideline compliance*). This addresses the first research question.
- **Planning-time evaluation.** Beyond compliance, we assess the intended actions described in maDMPs for their potential to enable FAIR practices (*planned FAIR alignment*), even when datasets do not yet exist or lack repository identifiers. This addresses the second research question.
- **Standards alignment.** Inputs consist of:
  - **maDMPs:** following the RDA DCS, ensuring compatibility with existing data management planning tools.
  - **FIP questions:** retrieved from community Nanopublication in *.trig* format, which are harvested and converted into internal catalogs of allowed values and patterns.

Outputs conform to the OSTrails FAIR Test Results specification, providing JSON-LD (and optional Turtle) representations for interoperability.

- **Evidence-first results.** Each test result is accompanied by the actual values observed, the mapping or benchmark it was checked against, and provenance metadata to support inspection and reproducibility.

This scope directly reflects the two research questions introduced in Chapter 1. Specifically, section 1.4

3.1.3 Use cases

This section outlines four concise, planning-time interactions with the evaluator. Each use case is written as a short scenario, kept at an abstract level; concrete component behavior is detailed later in Chapter 3.

UC1: Create mappings A facilitator selects a community FIP—either an existing profile or a newly harvested one from nanopublications—and aligns its questions with DCS fields. The outcome is a mapping that records which FIP creating questions are covered by which DCS fields. This output becomes the reference for subsequent checks and for reporting mapping coverage.

UC2: Check compliance A maintainer or reviewer provides a maDMP. The evaluator applies the active mapping and runs the three families of checks in one pass: completeness (presence and cardinality of required fields), feasibility (accuracy, availability, and internal consistency of planned actions), and guideline compliance (to check if the fields comply with the selected FIP, in the case they are present). Each item yields a pass, fail, or indeterminate outcome together with minimal evidence.

UC3: Produce an evaluation file A result of the evaluation is produced and readable for both human and machines. Results are grouped by benchmark groupings. This output highlights strengths and gaps and lists targeted recommendations linked back to the relevant FIP item and DCS path, so authors can revise the plan efficiently.

UC4: Enable semantic querying After the OSTrails-compliant JSON-LD file is created, it can be converted to Turtle. This allows semantic queries over the results (e.g., via SPARQL) and supports integration with external dashboards or registries discussed in Chapter 5.

Figure 3.2 summarizes how the three stakeholder roles relate to the four defined use cases. It shows that the Review System Facilitator configures mappings (UC1), while DMP Maintainers and Reviewers drive the evaluation activities (UC2–UC4).

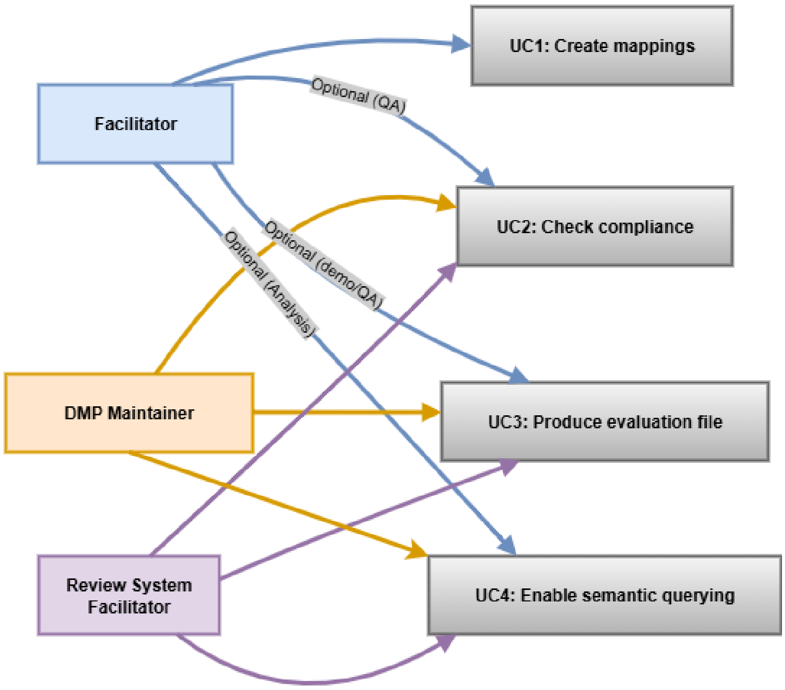

Figure 3.2: Stakeholder roles participation in the four use cases.

3.1.4 Defining Requirements

Requirements come from: stakeholders (Subsection 3.1.1), standards (DCS, FIPs, OS-Trails) and capabilities in our prototype.

We distinguish functional requirements FR (see Table 3.1) and non-functional requirements NFR (Table 3.2). the relation between Stakeholders, use cases and requirements is shown in Table 3.3.

Table 3.1: Functional requirements for the maDMP–FIP evaluator

ID	**Title**	**Description**
FR1	Load maDMP	Ingest a DCS maDMP and validate basic shape; paths and vocabularies.
FR2	Load FIP	Fetch the questions to create a FIP and the allowed values by the specific community.
FR3	Mapping Engine	Resolve FIP → DCS paths (1:n).
FR4	Completeness Tests	Check required entities/properties and cardinalities; return *present/not present/*.
FR5	Feasibility Tests	Check **accuracy** (identifier type, schema names), **availability** (URI/PID form), **consistency** (no contradictions).
FR6	Guideline Compliance	Enforce FIP constraints (value from maDMP compliant with allowed value from FIP?) on maDMP values.
FR7	Recommendations	Generate recommendations for non-compliant checks, linked to a specific FIP question.
FR8	Results as OSTRAILS	Export evaluation results as OSTrails JSON-LD with evidence and provenance.
FR9	TTL Conversion	Convert JSON-LD to *.ttl* for downstream SPARQL queries (presented in Chapter 5).
FR10	Aggregation Views	Compute per-Benchmark and per-goal results for reviewers and authors.

Table 3.2: Non-functional requirements.

ID	**Title**	**Description**
NFR1	Interoperability	Strict adherence to DCS/RDF inputs and OSTrails outputs; portable *.ttl*.
NFR2	Extensibility	Pluggable rules and profiles; new FIPs can be added without core changes.
NFR3	Traceability	Every result references the mapping, benchmark source and actual value.
NFR4	Determinism	Same input produces the same result.
NFR5	Usability	Clear summaries and recommendation phrasing for non-technical reviewers/authors.

Table 3.3: Stakeholders/Roles and their associated use cases and functional requirements

Stakeholders/Roles	Use case(s)	Functional requirement(s)
Review System Facilitator (research support staff, repository operators, infrastructure providers)	• **UC1: Create mappings** (Section 3.1.3)	• FR2 *Load FIP* • FR3 *Mapping Engine*
DMP Maintainer (researchers, data stewards)	• **UC2: Check compliance**	• FR1 *Load maDMP* • FR3 *Mapping Engine* • FR4 *Completeness Tests* • FR5 *Feasibility Tests* • FR6 *Guideline Compliance* • FR7 *Recommendations* • FR10 *Aggregation Views*
Reviewer (funders, ethics/legal, institutional admins, publishers)	• **UC2: Check compliance**	• FR1 *Load maDMP* • FR3 *Mapping Engine* • FR4 *Completeness Tests* • FR5 *Feasibility Tests* • FR6 *Guideline Compliance* • FR10 *Aggregation Views*
DMP Maintainer & **Reviewer**	• **UC3: Produce an evaluation file**	• FR8 *Results as OSTRAILS* • FR10 *Aggregation Views* • FR7 *Recommendations*
DMP Maintainer, Reviewer, Review System Facilitator	• **UC4: Enable semantic querying**	• FR9 *TTL Conversion* • FR8 *Results as OSTRAILS* (pre-requisite)
Review System Facilitator	• **UC1: Create mappings** • **UC3: Produce an evaluation file** (configure outputs, evidence/provenance)	• FR8 *Results as OSTRAILS* (schema conformance) • FR10 *Aggregation Views* (configure benchmark groupings and summaries)
All roles	• Cross-cutting across **UC2–UC4** (review, reporting, feedback)	• FR7 *Recommendations* (actionable guidance) • FR8–FR10 (interoperable outputs, TTL conversion, summaries)

3.2 Architectural Objectives and Constraints

This architecture operationalizes the requirements derived in Section 3.1 (Tables 3.1–3.2) and prepares for the implementation (Chapter 4). Objectives 1–5 are derived from NFRs:

- **Objective 1 (Interoperability).** Ingest maDMPs as DCS JSON (optionally DCSO RDF) and emit **OSTRAILS JSON-LD**; enabling possible conversion to RDF for SPARQL.
- **Objective 2 (Rule-based evaluation).** Execute mapping-driven tests for completeness, feasibility (accuracy, availability, consistency), and guideline compliance, producing *Pass*/*Fail*/*Indeterminate* outcomes.
- **Objective 3 (Evidence & provenance).** Attach values used for the evaluation, mapping/benchmark references, etc.
- **Objective 4 (Extensibility).** Add new FIPs to enable the possibility of correctly evaluating more maDMPs
- **Objective 5 (Determinism).** Produce stable outputs.

Scope. The system evaluates the contents of a maDMPs against community FIPs (planning-time alignment). Dataset-level FAIR scoring by tools such as F-UJI is out of scope for the core evaluator and only discussed for context in Chapter 5.

3.3 Conceptual Framework

This work conceptualizes automated maDMP evaluation as the transformation of a DCS-compliant maDMP and a selected FIP into evidence-bearing, OSTrails-compliant FAIR Test Results. Figure 3.3 show the overview for the proposed solution and figure 3.5 ilustrates the process more in detail. The framework distinguishes three layers and one mapping status taxonomy [GW+25]:

Metrics (FIP questions). Each of the 21 questions required to create a FIP are modeled as a *metric* addressing a FAIR aspect (e.g., identifier type, Data usage license, Communication protocol). Metrics define *what* is assessed.

Tests (compliance). For each metric, one or more *tests* compare observed maDMP values against allowed values from the FIP mapping and constraint catalogs, yielding *Pass* or *Fail*. When comparison is not possible due to the community not having specified an allowed value for a FIP question, the outcome is *Indeterminate*. Tests record both the status (*prov:value*) and the observed evidence (*ftr:log*).

Benchmarks (metric groupings). Benchmarks are *groupings of metrics* reported together for interpretability (e.g., the pair of metrics for FAIR subprinciple A1.1). Benchmarks express a community's perspective on what FAIR means in practice. *Note:* allowed value lists (licenses, PID schemes, metadata standards) are *not* part of benchmarks; they live in *Constraint Catalogs* and are referenced by tests.

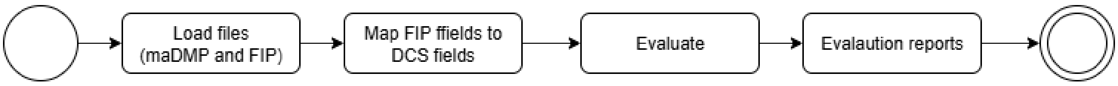

Figure 3.3: Overview workflow for the proposed solution.

3.3.1 Mapping FIP Questions to DCS

A key step in our methodology is the alignment of FIP questions with DCS fields. This establishes the bridge between community expectations and the contents present in the maDMP. Each FIP item (e.g., "Which usage license do you use?") is linked to one or more DCS paths (e.g., *dmp.dataset.distribution.license.license_ref*) as ilustrated in figure 3.4. The implementation details such as how controlled vocabularies are built and validated are covered in Chapter 4.

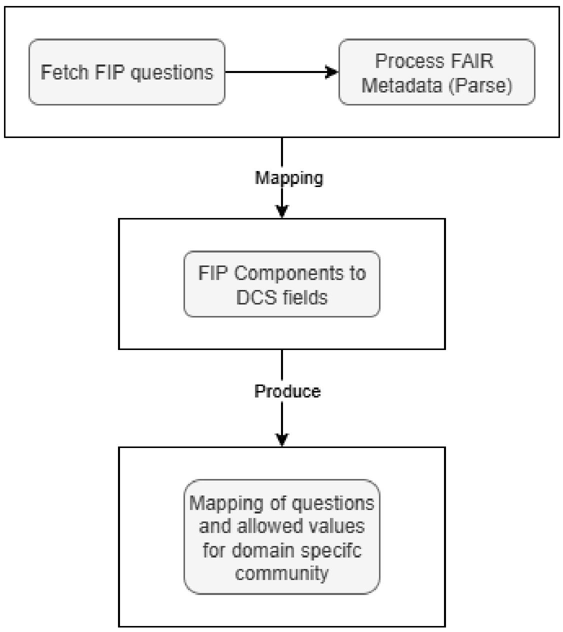

Figure 3.4: Conceptual mapping from FIP questions to DCS fields.

3.3.2 Metrics, Tests, and Benchmarks

Once mappings are established, we can define metrics that represent evaluation goals, tests that operationalize them, and benchmarks that group related metrics under FAIR sub-principles (following the concepts described at the beginning of Chapter 3). For instance, the Identifier type Benchmark could group both subprinciples from FAIR principle F1. Another example could be feasibility tests verifying the accuracy and consistency of the fields maDMP and guideline compliance tests checking whether the values are in line with FIP expectations.

3.4 Detailed Pipeline

As it can be seen in Figure 3.5, the pipeline of the proposed solutions consists of:

- **(Optional) Nanopub Harvester.** When available, retrieves FIP questions from FIP Nanopublications to build a list of FIP questions and allowed values. It could not be necessary if the desired FIP is already listed.
- **Select FIP** Selects FIP from desired domain.
- **FIP→DCS Mapping Engine.** Binds FIP questions to DCS paths (1:n); One FIP question could relate to more than 1 fields from the DCS (e.g., R1.1 could be found under license_name or license_ref). The mapping status can be *Mapped*, *Partially Mapped*, or *Indeterminate*.
- **Load maDMP** Which is evaluated against the selected FIP.
- **Test Engine.** Executes tests (*Pass/Fail/Indeterminate*) and groups metrics into Benchmarks(for the JSON-LD output) . Checks Completeness and Feasibility (accuracy/availability/consistency).
- **Evidence & Provenance Collector.** Records maDMP values, allowed values from the selected FIP, status and other information required by the test result vocabulary [GW+25].
- **OSTRAILS Serializer.** Emits OSTrails-compliant JSON-LD assessment results.
- **TTL Converter (Optional)** Converts JSON-LD to TTL format for possibility of semantin querying. See section 5.5
- **Produces Reports.** OSTrails compliant evaluation report, compliance table, metadata validation, goals checks and recommendations for non-compliant fields.

3.4.1 FIP questions to DCS fields mapping

Each FIP question is mapped to one or more DCS paths in maDMPs. Mappings are classified as:

- **Mapped:** direct DCS path(s) satisfy the FIP question (e.g., *dataset.dataset_id.identifier* for “What globally unique, persistent, resolvable identifiers do you use for datasets?”).

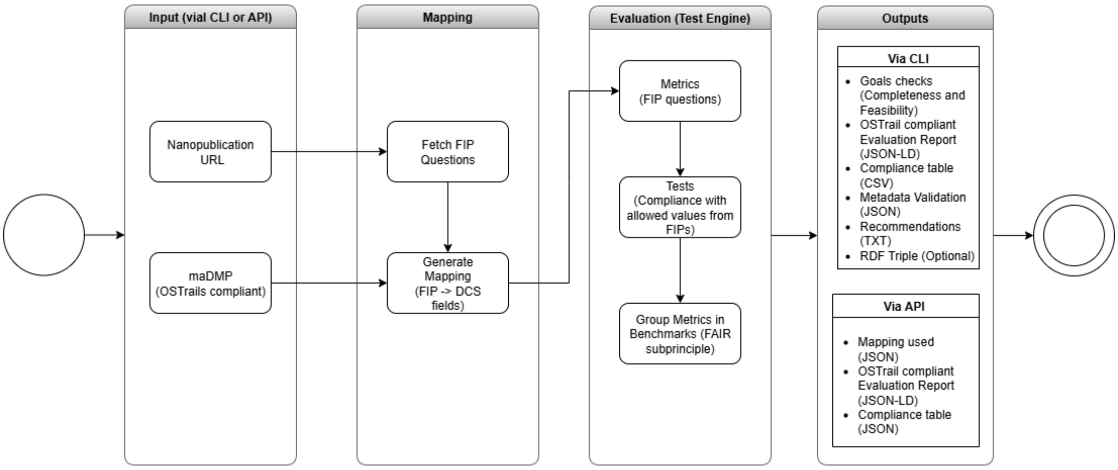

Figure 3.5: Framework of the proposed solution.

- **Partially Mapped:** DCS captures only the abstract answer (e.g., a high-level access category (open or closed) but not the specific authentication/authorization method expected by A1.2).
- **Not Mapped:** no reliable mapping exists or the value cannot be safely classified from the available fields.

This process goes through expert review before use in evaluation. This ensures reproducibility and determinism. The complete mapping can be seen in Figure 3.6

3.4.2 Evaluation Method

Finally, we outline the evaluation procedure that will later be applied in Chapter 5. The protocol consists of four broad steps:

1. **Input**: select a FIP and provide a maDMP.
2. **Mapping**: align FIP questions to DCS fields.
3. **Testing**: run completeness, feasibility, and compliance checks.
4. **Output**: export structured results for later analysis.

3.5 Test Engine

This section details the *Test Engine* component shown in Figure 3.5. The engine determines which metrics are testable from the current FIP to DCS, metrics without a mapped DCS path are marked as indeterminate. It also computes compliance by comparing observed maDMP values in the mapped DCS paths against the community

FAIR principle	FIP question	maDMP field	Mapping status
F1	What globally unique, persistent, resolvable identifiers do you use for metadata records?	dataset.dataset_id.identifier	Mapped
F1	What globally unique, persistent, resolvable identifiers do you use for datasets?	dataset.dataset_id.identifier	Mapped
F2	Which metadata schemas do you use for findability?	dataset.metadata.metadata_standard_id.identifier	Mapped
F3	What is the technology that links the persistent identifiers of your data to the metadata description?	dataset.distribution.host.pid_system	Mapped
F4	In which search engines are your metadata records indexed?	dataset.distribution.access_url	Mapped
F4	In which search engines are your datasets indexed?	dataset.distribution.access_url	Mapped
A1.1	Which standardized communication protocol do you use for metadata records?	dataset.distribution.host.url	Mapped
A1.1	Which standardized communication protocol do you use for datasets?	dataset.distribution.host.url	Mapped
A1.2	Which authentication & authorisation technique do you use for metadata records?	dataset.distribution.data_access	Partially Mapped
A1.2	Which authentication & authorisation technique do you use for datasets?	dataset.distribution.data_access	Partially Mapped
A2	Which metadata longevity plan do you use?		Not Mapped
I1	Which knowledge representation languages (allowing machine interoperation) do you use for metadata records?		Not Mapped
I1	Which knowledge representation languages (allowing machine interoperation) do you use for datasets?		Not Mapped
I2	Which structured vocabularies do you use to annotate your metadata records?	dataset.metadata.metadata_standard_id.identifier	Partially Mapped
I2	Which structured vocabularies do you use to encode your datasets?	dataset.metadata.metadata_standard_id.identifier	Partially Mapped
I3	Which models, schema(s) do you use for your metadata records?	dataset.metadata.metadata_standard_id.type	Mapped
I3	Which models, schema(s) do you use for your datasets?	dataset.metadata.metadata_standard_id.type	Mapped
R1.1	Which usage license do you use for your metadata records?	dataset.distribution.license.license_ref	Mapped
R1.1	Which usage license do you use for your datasets?	dataset.distribution.license.license_ref	Mapped
R1.2	Which metadata schemas do you use for describing the provenance of your metadata records?		Not Mapped
R1.2	Which metadata schemas do you use for describing the provenance of your datasets?		Not Mapped

Figure 3.6: Mapping from FIP questions to RDA DCS maDMP fields, with a status per FAIR principle (*Mapped*, *Partially Mapped*, *Not Mapped*).

expectations from the FIP. Finally, it aggregates metric results into benchmarks. All checks are bound to explicit mapping entries and to the constraint catalogs (e.g., identifier types, licenses, metadata standards):

- **Completeness (RDA DMP Common Standard v1.2).** Presence and basic cardinality checks against the DCS v1.2 schema[RDA25]. We distinguish *always-required* fields (e.g., *dmp_id.identifier/type*, *title* and *conditional* fields required only when certain fields are present (e.g., if *host* exists, require *host.title* and *host.url*.
- **Feasibility.**
  - **Accuracy:** lightweight syntactic plausibility checks (e.g., URL format for dataset identifiers, host URLs, and license URLs).
  - **Availability:** optional online resolvability tests for identifiers/URLs (e.g., HEAD requests).

  - **Consistency:** cross-field predicates (e.g., *data_access=open* should be accompanied by a license and distributions should declare *byte_size*.

- **Guideline compliance (FIP-driven).** This is a main focus of the system. For each of the 21 FIP questions (treated as *metrics*), tests compare the observed maDMP values at the mapped DCS paths against the allowed sets/patterns from the *constraint catalogs*. Each test yields *Pass/Fail/Indeterminate*, while the conceptual evidence model distinguishes *field status* (present/not present) and *compliance category* (compliant, non-compliant, missing value, not applicable). Results capture provenance to justify each decision.

3.5.1 Evaluation of early maDMP stages

In early research stages, when a maDMP has not yet produced datasets (initial phase of the research), researchers may include a reference to an *existing, representative dataset* (e.g., a DOI/URI to a similar dataset).This would give the researcher(s) an idea of how close their intended actions align with the FIP they are evaluating against. This proposed solution aligns with the proposed strategies by Miksza et al. [TP23].

3.6 Alignment with the OSTrails FAIR Test Results vocabulary

We adopt the OSTrails FAIR Test Results (FTR) model for result interoperability and specify here the design conformance profile that our tool enforces for the core FTR classes. The concrete implementation details are provided in Chapter 4. We conform to the OSTrails FAIR Testing Resource Vocabulary (see Figure 3.7).

Table 3.4 details the required properties, relations, and subclasses that our tool guarantees for the seven main classes.

Table 3.4: Required fields of the OSTrails FAIR Test results main concepts. (Part 1)

Class	Required metadata	Properties & subclasses
ftr:Test	• **dcterms:identifier** (xsd:string) • **dcterms:title** (xsd:string) • **dcterms:description** (xsd:string) • **dcterms:license** (xsd:anyURI) • **dcat:version** (xsd:string)	• **sio:is-implementation-of** → `dqv:Metric` • *subClassOf* `dcat:DataService`, `prov:Agent`
ftr:TestResult	• **dcterms:identifier** (xsd:string) • **dcterms:title** (xsd:string) • **dcterms:description** (xsd:string) • **dcterms:license** (xsd:anyURI) • **prov:value** (xsd:string) *[pass?]* • **ftr:log** (xsd:string) *[allowed value]*	• **ftr:outputFromTest** → `ftr:Test` • **prov:wasDerivedFrom** → `prov:Entity` • *subClassOf* `prov:Entity`
ftr:TestResultSet	• **dcterms:identifier** (xsd:string) • **dcterms:title** (xsd:string) • **dcterms:license** (xsd:anyURI)	• **prov:hadMember** → `ftr:TestResult` • **prov:wasDerivedFrom** → `prov:Entity` • **prov:wasGeneratedBy** → `ftr:TestExecution\ \Activity` • *subClassOf* `prov:Entity`, `prov:Collection`
ftr:Benchmark	• **dcterms:identifier** (xsd:string) • **dcterms:title** (xsd:string) • **dcterms:description** (xsd:string) • **dcat:version** (xsd:string)	• **ftr: hasAssociatedMetric** → `dqv:Metric`

continued on next page

Table 3.4: Required fields of the OSTrails FAIR Test results main concepts. (Part 2)

Class	Required metadata	Properties & subclasses
ftr:Algorithm	• **dcterms:identifier** (xsd:string) • **dcterms:title** (xsd:string) • **dcterms:description** (xsd:string) • **doap:repository** (xsd:anyURI) • **dcterms:license** (xsd:anyURI) • **dcat:version** (xsd:string)	• **sio:is-implementation-of** → `ftr:Benchmark` • **dcterms:creator** / **vcard:contactPoint** → `vcard:Individual` **or** `vcard:Organization` • *subClassOf* `dcat:DataService`, `prov:Agent`
dqv:Metric	• **dcterms:identifier** (xsd:string) • **dcterms:title** (xsd:string) • **dcterms:description** (xsd:string) • **dcat:version** (xsd:string)	• *Referenced from* `ftr:Test` via `sio:is-implementation-of` • *Associated with* `ftr:Benchmark` via `ftr:hasAssociatedMetric`
ftr:TestExecutionActivity	• *(no additional literals required in our profile)*	• **prov:wasAssociatedWith** → `ftr:Test` • **prov:generated** → `ftr:TestResult` • **prov:used** → `prov:Entity` • *subClassOf* `prov:Activity`

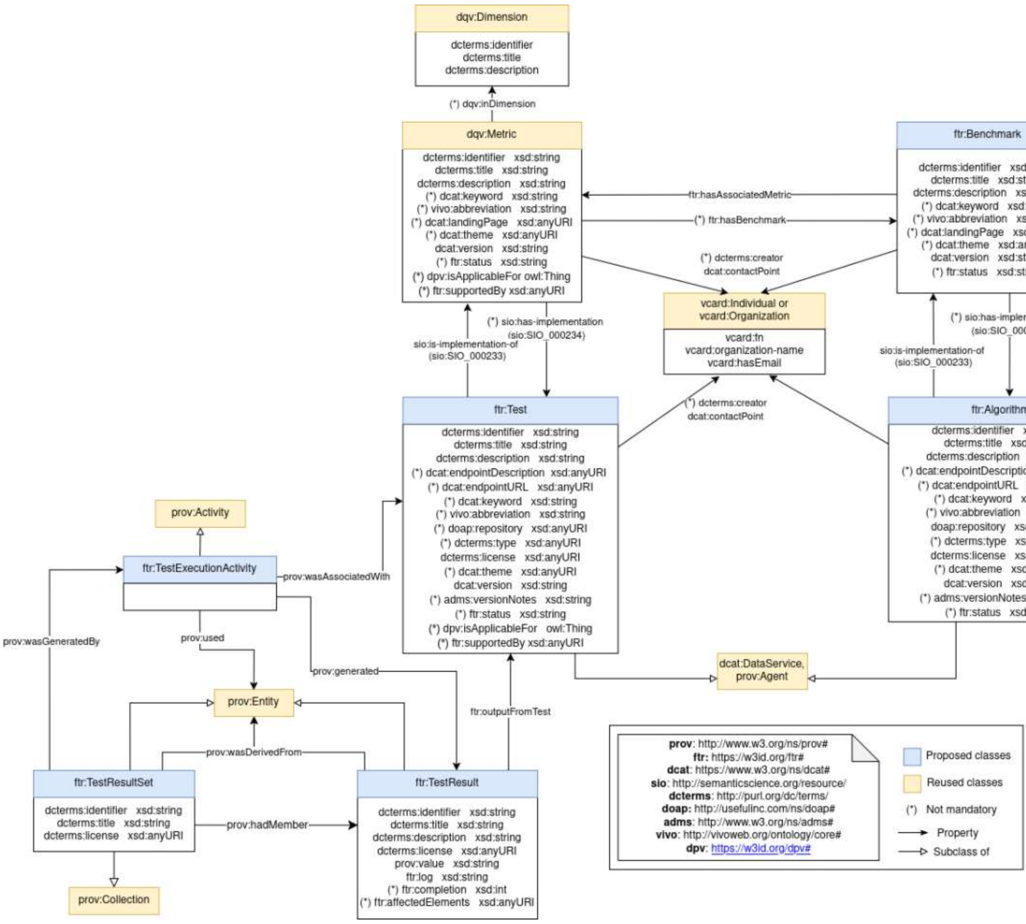

Figure 3.7: OSTrails FAIR Test Results (FTR) vocabulary concepts and relations used by the system (We focus on the main concepts discussed in section 3.6). This figure illustrates the serialization target referenced throughout Section 3.6. [GW+25]

3.6.1 Results overview for OSTrails FTR

We previously mentioned and defined the main concepts in the OSTrails FTR in chapter 2, subsection 2.5.1 [GW+25]. Besides each *prov:value* in a TestResult (*Pass*/*Fail*/*Indeterminate*), each *ftr:TestResult* also provides the following two dimensions:

- **Field status:** whether the maDMP contains a value at the mapped DCS path (*Present* / *Not Present*).
- **Compliance:** the relation of the observed value (if any) to the allowed value supplied by the FIP mapping:
  - *Compliant* – a present value matches the allowed values,
  - *Non-compliant* – a present value does not comply with the allowed values,

– *Missing value* – a mapped DCS field is empty in the maDMP,
– *Not Applicable* – no DCS path or no allowed values are provided by the mapping.

In our evaluation, the result (*Pass/Fail/Indeterminate*) is stored under *prov:value*, while the *field status* and *compliance* are attached as human-readable evidence in *dcterms:description*. Both the *ftr:Test* and the *ftr:TestResult* include such descriptions to provide human-readable explanations for provenance. The associated *ftr:Test* carries the metadata of the evaluated field (DCS path and allowed value). This ensures that even in cases where compliance cannot be judged, the reason (*missing value* vs. *not applicable*) is explicit at design time.

3.6.2 Grouped metrics (Benchmarks)

As discussed previously in subsection 2.5.1, Benchmarks are collections of metrics. In our framework, we define twelve benchmarks, each corresponding to a related group of FAIR sub-principles. These benchmarks structure the evaluation and provide a narrative view of compliance:

- F1: Identifier type
- F2: Metadata schema
- F3: Metadata–Data linking mechanism
- F4: Search Engines
- A1.1: Communication protocol
- A1.2: Authentication and authorization technique
- A2: Metadata longevity
- I1: Knowledge representation language
- I2: Structured vocabularies
- I3: Metadata/data schema
- R1.1: Data usage license
- R1.2: Provenance model

3.6.3 Quality Assurance

The set of FIP questions is harvested *automatically* from the FIP nanopublications. Because no manual work is involved, evaluations are deterministic.

3.6.4 Interfaces

Two modalities are created to support execution (details in Chapter 4):

- **CLI.** To run the tool directly from the command line.
- **API.** To enable integration with external services, projects, or tools.

3.7 Summary

This chapter described the methodological foundations and the conceptual system design for automated maDMP evaluation. In Section 3.1 we identified who interacts with DMPs and what they need, derived use cases, and stated functional and non-functional requirements that drive the solution It introduced the mappings as well as the definition of metrics, tests, and benchmarks that form the evaluation approach. The chapter then showed how community expectations expressed in FIPs can be aligned with DCS-based maDMP and assessed through structured results in the OSTrails format.

Building on this, the system architecture was outlined, including the main components, their responsibilities, and the evaluation pipeline. Key elements such as FIP ingestion (from Nanopublication), expert-validated FIP to DCS mappings, constraint catalogs, the Test Engine that produces Pass/Fail/Indeterminate outcomes, and the OSTrails-aligned result model were presented. Concrete implementation details are left for Chapter 4. Results based on this conceptual design are reported in Chapter 5.

CHAPTER 4

Implementation

This chapter details the concrete implementation of the maDMP–FIP evaluation framework described in Chapter 3. We explain the internals of the Nanopublication-to-mapping pipeline, how the questions are mapped and ordered, how controlled vocabularies and validation rules are applied, the evaluator's decision semantics and outputs, the OSTrails FAIR Test Results (FTR) serialization [GO25, GW⁺25], and the CLI/API interfaces.

4.1 Module Interaction Overview

In this section, we describe *how* the implemented components collaborate at run time. This builds on the workflow defined in Section 3.4, Figure 3.5:

Runtime roles.

a) **Ingestion**: parses a DCS-compliant maDMP and provides a model with structured dot-path access to nested entities sucha as datasets, distributions and contributors. It does not alter content; it simply makes available the values and types required for subsequent steps in the workflow.

b) **Mapping Manager**: loads the FIP→maDMP mapping (harvested from Nanopublications), uses a lightweight *Question Map* to attach to each question URI its FAIR code, the canonical question text, and the corresponding DCS selector(s), and then enforces a stable display order (F→A→I→R, followed by the sub-principle).

c) **Constraint Catalogue Service**: supplies versioned allowed-value sources (e.g., SPDX licenses list for licenses) and curated/domain lists or patterns elsewhere (regular expressions).

d) **Evaluation Engine**: executes *tests* per metric (the 21 FIP questions) (completeness, feasibility and guideline compliance). It returns a *field status* (Present/Not Present), a *compliance category* (Compliant/Non-compliant/Missing value/Not Applicable), and a canonical decision (Pass/Fail/Indeterminate).

e) **Evidence Recorder**: assembles human-readable evidence (observed value, status, category) and traces which selector and catalogue reference were used, so results are auditable and deterministic.

f) **Result Serializer**: materializes the outcome as OSTrails FTR (JSON-LD), linking Metrics, Tests, TestResults, Benchmarks, and the run-level Activity/ResultSet; optional TTL is produced (e.g., for SPARQL queries).

g) **Interface Layer**: a CLI and a web API drive the same flow and additionally allow *uploading/registering a new mapping* from a nanopublication URL.

4.2 Nanopublications to a FIP-DCS Mapping

This section explains how a FIP Nanopublication is transformed into a mapping that the evaluator can use. The implementation steps ar described in the following subsections. Figure 4.1 shows the detailed flow chart.

4.2.1 Nanopublication retrieval and parsing

The harvester requests the provided URI with a content negotiation header that prefers TriG format. It then attempts to parse the returned payload *in order*: *trig* → *nquads* → *xml* (RDF/XML) → *json-ld* (Table 4.1 describes each of them). If parsing fails, the execution is canceled.

Table 4.1: Nanopublication parsing supported formats used by the harvester.

Serialization tried	**Description**
TriG (*application/trig*)	Default registry format with named graphs.
N-Quads (*application/n-quads*)	Quad-based fallback if TriG fails.
RDF/XML (*application/rdf+xml*)	Legacy XML-based RDF format.
JSON-LD (*application/ld+json*)	JSON syntax for Linked Data.

4.2.2 Locating the declaration index and enumerating declarations

Given a FIP Nanopublication, the harvester finds the declaration index via the predicate *has-declaration-index* and retrieves its content. The index lists are stored under

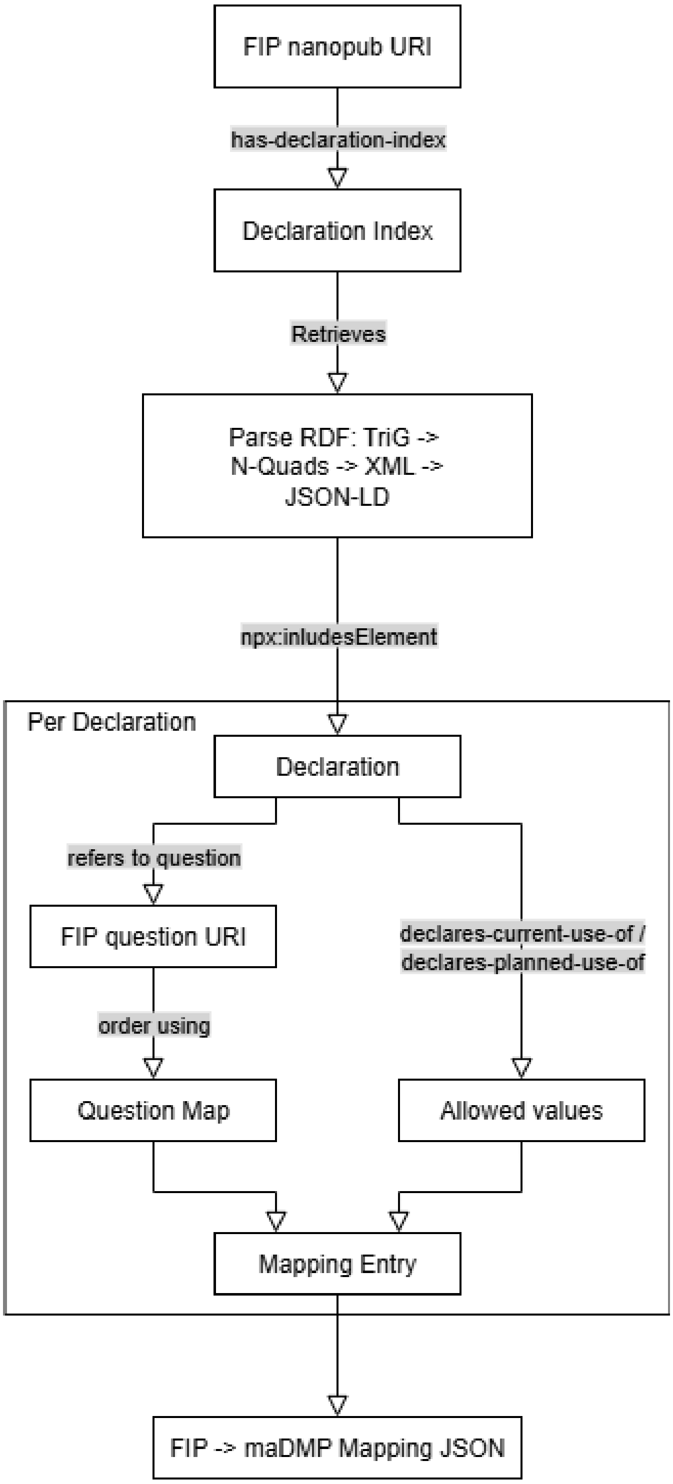

Figure 4.1: Flowchart for creating a new FIP-to-maDMP mapping, applied once per FIP question.

npx:includesElement. The harvester iterates these URIs, processes each declaration, and accumulates the answers per question.

4.2.3 Processing each declaration

A single declaration refers to exactly one FIP question (via *refers-to-question*) and provides *allowed values* by the community. The harvester:

a) resolves the question URI and looks up metadata in the *Question Map* (Section 4.2.4), obtaining the FAIR sub-principle code (F1-R1.2) and the canonical question text.

b) extracts all values from *declares-current-use-of* and *declares-planned-use-of*.

c) extracts any considerations (free-text comments) and a version if provided via Schema.org;

d) returns a record with: *Question_URI*, *FAIR_principle*, *FIP_question*, *DCS_field*, *Mapping_status*, *Comments* and *Allowed_values*.

If multiple declarations are present for the same question (e.g., several values), their *Allowed_values* are *merged* (using a "|" separator), avoiding duplicated values while preserving order. The harvester also examines the FIP node for a human-readable label (*rdfs:label*) to mint a filesystem-safe mapping filename.

4.2.4 Question Map and stable ordering

The Question Map establishes, for each canonical FIP question URI, the FAIR code (e.g., F1, A1.1), the dot-path to inspect and the canonical question text. It guarantees a stable ordering of questions in the final mapping: all 21 URIs are listed in FAIR sequence (from 1 through R1.2), as defined in Table 2.3[GO nd]. This works even if no allowed values are defined for a given question. This ensures that reports always display the questions in a consistent FAIR order, independent of which values are present in a given maDMP and supporting human-readability. See Table 4.2 for an example of a mapping between a FIP question, and it's relevant DCS field.

Table 4.2: Conceptual schema of a Question Map entry.

Field	**Example**	**Meaning**
Question_URI	*https://w3id.org/fair/fip/terms/FIP-Question-A1.1-D*	Canonical metric identifier (FIP question URI).
principle	A1.1	FAIR sub-principle (used for grouping/benchmarks).
madmp	*dataset.distribution.host.url*	dot-path field.
text	Which standardized communication protocol do you use for datasets?	Canonical question text.

4.2.5 Mapping materialization

The harvester produces a JSON document with the following structure:

Listing 4.1: Example FIP to maDMP mapping structure

```
{
  "FIP_Version": "",
  "FIP_maDMP_Mapping": [
    {
      "Question_URI": "",
      "FAIR_principle": "",
      "FIP_question": "",
      "DCS_field": "",
      "Mapping_status": "",
      "Comments": "",
      "Allowed_values": []
    },
  ]
}
```

where each array element is one mapped FIP question with the semantics in Section 4.2.3. A utility function transforms this list into a dictionary keyed by question label/URI for efficient lookups during evaluation, keeping *Allowed_values*, *DCS_field*, *Mapping_status*, and the FAIR subprinciple identifier.

4.3 Controlled vocabularies and Validation Rules

We check *dataset.distribution.license.license_ref* (metric R1.1. See subsection 3.6.2) against the official SPDX licenses license list [Lin23]. At runtime, the checker downloads the SPDX JSON, then normalizes the input URL and each SPDX URL (lowercases scheme/host, trims trailing slashes and stripping). If any normalized input URL matches an SPDX licenses (present in allowed_values), the result is *True* (compliant); otherwise *False* (non-compliant).

4.3.1 Domain-specific questions.

Other questions (e.g., "In which search engines are your datasets indexed?") often have *community-specific* values (e.g., https://globalbioticinteractions.org/worldfair); there is no universal registry. The implementation for these cases supports regular Expressions (regex) for common identifier or service patterns and exact matching against curated known labels (e.g., Schema.org, DCAT, Dublin Core). If no match is found, the value is reported as Unknown.

4.3.2 Consistency predicates.

Cross-field rules capture the consistency of the maDMP. For example, open access implies the presence of a license; personal vs. sensitive data flags must not be contradictory; and distributions should declare basic integrity properties (e.g., byte size). Issues are reported per dataset/distribution.

4.4 Evaluator Core and Decision Semantics

4.4.1 Path resolution and value collection

The evaluator accepts a DCS-compliant maDMP in JSON format and collects all values that match a dot-path selector. The path walker descends through dictionaries/lists, yielding each concrete value; this supports multi-valued fields (e.g., multiple distributions or licenses).

4.4.2 Per-metric evaluation

For each metric (the 21 FIP questions, in the stable order from the Question Map), the evaluator:

1. resolves the mapped DCS path(s) and computes a Field status (*Present*/*Not Present*);
2. applies the appropriate test(s): presence (completeness), membership/pattern (accuracy), and cross-field consistency (consistency);
3. assigns a compliance category: *Compliant*, *Non-compliant*, *Missing value* (mapped but absent), or *Not Applicable* (no mapped path or no allowed values);
4. emits the canonical decision (*Pass*/*Fail*/*Indeterminate*) and per-value compliance list when multiple values were found (prov:value).

Table 4.3 shows this more in detail.

Table 4.3: Conceptual decision matrix (mapping context × field status).

Mapping context	**Field status**	**Compliance category**	**Result (prov:value)**
Path mapped & allowed values provided; maDMP value *not provided*	Not Present	Missing value	Fail
Path not mapped or no allowed values provided	Present/Not Present	Not Applicable	Indeterminate
Path mapped; value in allowed set/pattern	Present	Compliant	Pass
Path mapped; value not in allowed set/pattern	Present	Non-compliant	Fail

4.4.3 Human-readable summaries and recommendations

The evaluator produces two human-facing artifacts: a *compliance table* that records, for each question, the mapped DCS field, observed values, allowed values, and a per-value "Yes/No" result for compliance; and a concise *recommendations* file that lists missing or non-compliant items. If a community has made *no choice* (no allowed values), the table explicitly notes this so reviewers understand why compliance could not be evaluated.

4.4.4 Goal checks and metadata validation

Beyond FIP compliance, the tool computes goal-oriented indicators (completeness, accuracy, availability, consistency) and a metadata intent validation bundle. Availability probes use lightweight HEAD request requests to check resolvability of identifiers and URLs; these do *not* alter FTR results but are reported as separate JSON artifacts for possible reviewers.

4.5 Serialization to OSTrails FTR

The prototype emits a compact JSON-LD document where each FIP question is a *dqv:Metric*, implemented by a single *ftr:Test*, which yields one *ftr:TestResult*. *ftr:Benchmarks* are created per FAIR sub-principle (e.g., both question from *A1.1*) and list their associated metrics; the evaluation *ftr:Algorithm* node declares that it implements these benchmarks (for discoverability). Each *ftr:Test* description records the DCS dot-path and the set of allowed values (if any) used for the decision; each *ftr:TestResult* carries the canonical result in *prov:value* and a human-readable *dcterms:description* with evidence.

Finally, an *ftr:TestResultSet* collects all *ftr:TestResults* from a single run. The run itself is captured as a *ftr:TestExecutionActivity*, i.e., the action performed by an agent (such as invoking the API) that generated the TestResult or TestResultSet and links them back to the evaluated maDMP. Figure 4.2) focuses on the seven core classes. The full FTR model can be seen in Figure 3.7.

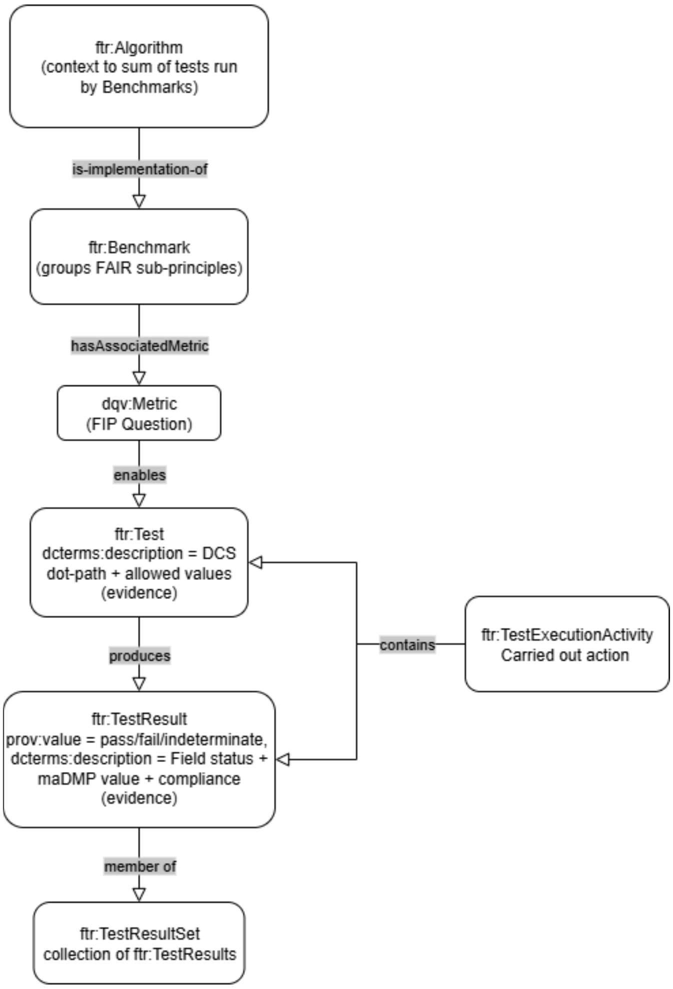

Figure 4.2: Formatting flowchart. From evaluator artifacts to FTR entities.

4.6 Interfaces: CLI and API

Running via the CLI . The CLI entry point loads the maDMP and the selected mapping, executes evaluation, prints a present/compliant summary, and writes (with a single command):

- Compliance table (CSV)

- Recommendations file (TXT)
- goals_check for completeness and feasibility (JSON)
- metadata_validation (JSON)
- FTR JSON-LD

Additionally, there´s the possibility to create new mappings. Also, the the output file can be converted to a RDF triple representation (TTL) for semantic querying.

Running via API endpoints. The service exposes:

- **upload_fip** Accepts a nanopublication URL, harvests and stores a mapping (also updates the UI dropdown)
- **evaluate** Accepts a maDMP JSON and a selected mapping, returning the mapping used (JSON), the OSTrails-compliant JSON-LD result, and the compliance table (JSON)

4.7 Runtime Walkthrough

This walkthrough explains the steps an evaluation goes trough. The empirical results are given in Chapter 5.

1. **Prepare context.** A mapping is selected (already present or harvested from a FIP nanopublication). The Mapping Manager builds the Question Map (FAIR code, canonical question text, DCS dot-path) and fixes a stable order (F → A → I → R, then sub-principle).
2. **Load the plan.** The maDMP is parsed as DCS JSON and exposed through a dot-path accessor so nested entities (datasets, distributions, licenses, . . .) can be queried uniformly.
3. **Evaluate the 21 metrics.** For each FIP question in Question-Map order: resolve the mapped dot-path, compute field status (Present/Not Present), apply the appropriate strategy (presence, membership/pattern, consistency), assign a compliance category (Compliant/Non-compliant/Missing value/Not Applicable), and set the canonical decision (*Pass*/*Fail*/*Indeterminate*). Per-value compliance is recorded when multiple values occur.
4. **Aggregate goals and validate intents.** Completeness, accuracy, availability, and consistency indicators are provider in both human and manchine-readable formats for reviewers.

5. **Serialize and export.** Results are materialized as OSTrails FTR in JSON-LD (optionally also TTL). Reviewer-friendly outputs (compliance table and recommendations) are written alongside.

6. **(Optional) Explore.** The serialized graph could be queried with SPARQL to inspect benchmarks, metrics, and evidence (in case the TTL RDF instance was created.

4.8 Boundaries and considerations

The core FIP-compliance evaluator is deterministic and operates on local inputs (FIP mapping + maDMP). This means the evaluation itself runs locally (on the host executing the call); networking occurs only for the following:

- **Nanopublication ingestion** To add the mapping of a new FIP.
- **Catalogue refresh for SPDX licenses** To obtain the latest license list from the Linux Foundation SPDX Workgroup[Lin23].
- **Availability checks** to make a HEAD request request (resolvability of URLs/identifiers).

4.8.1 Performance and Limitations

In what about complexity refers, the evaluation is linear in the number of mapping entries and maDMP entities; the current implementation evaluates one maDMP at a time, requiring multiple runs to evaluate multiple objects.

Limitations include, for example, that when using regular expression-based checks the tool might be too permissive. For instance, if the allowed value for the question "In which search engines are your datasets indexed?" is the GBIF search engine, and a regex for globi is defined, the tool might consider the value *https://www.globalbioticinteractions.org/?interactionType=pollinates* as compliant, even if the link is not accessible at the moment. This does not mean that the tool is incorrect: it correctly checks compliance against the declared FIP, but temporary unavailability of a site does not imply that the maDMP value itself is non-compliant. As noted earlier in Subsection 4.3.1, regex support helps cover community-specific cases, but it may not guarantee accessibility. Additional outputs such as metadata validation or goal checks can help reviewers detect such situations. This is further discussed in Chapter 6.

4.9 Summary

This chapter presented the concrete implementation of the evaluation framework described in Chapter 3. The system ingests maDMPs together with community FIP mappings (curated or harvested from nanopublications) and evaluates each of the 21 FIP questions in a stable order defined by a Question Map. For every metric, the evaluator determines whether the relevant field is present, how the observed values relate to allowed catalogues or patterns, and records both a canonical decision and human-readable evidence.

Constraint information is managed explicitly: for example, license validation is grounded in the official SPDX licenses catalogue, while more domain-specific questions fall back on curated lists or conservative pattern checks. Network calls are only made in specific cases such as nanopublication harvesting, refreshing the SPDX licenses license list, or performing availability tests. In all other cases, the compliance checks run locally and remain fully deterministic. Provenance information records the mapping version, values found in both maDMPs and FIPs, etc. This ensures reproducibility within each run.

The implementation produces results in the form of OSTrails FAIR Test Results, serialized as JSON-LD and optionally convertible to TTL for semantic queries. The evaluation pipeline is accessible through both a lightweight CLI and a hosted API, each exposing the same core functionality and allowing the registration of new mappings. Overall, the design balances automation and transparency: compliance checks are simple, explainable, and evidence-bearing.

In the next chapter (Chapter 5), we demonstrate how this implementation performs in practice when applied to real maDMPs, highlighting its coverage, reproducibility, and the added value of community-defined FIPs for planning-time assessment.

CHAPTER 5

Evaluation and Results

In this chapter, we present and discuss the results obtained from applying the implementation detailed in Chapter 4. We run the evaluator using real published maDMPs and FIPs to see how the system behaves in practice. We start by describing the setup and inputs that were used, then move on to show the mapping coverage and the outcomes of the planned FAIR alignment checks. We also discuss a experiment where we a representative link to a dataset could be added for early-stage DMPs to see how intended actions could be evaluated. Finally, we show some SPARQL-based explorations over the OSTrails FTR exports and check how well the tool satisfied the requirements and use cases defined earlier.

To complement these planning-time results, we also include an illustrative FAIRness evaluation using F-UJI. This gives a quick view of how a FAIRness evaluation using a FER automated tool looks like.

5.1 Experimental Setup

Case study selection. To put the protoype evaluation tool into practice, we selected two publicly available maDMPs and evaluated each to a domain specific FIP:

- **Case Study 1 (Climate)**: *IAM COMPACT – Machine-Actionable DMP (New)* maDMP evaluated against the *F2A Case Study 6: FAIR Climate Risk Assessments* FIP. The IAM COMPACT project (Horizon Europe) uses ensembles of integrated assessment models to support assessment of global climate goals and the next round of NDCs; the open maDMP (maintained in ARGOS) documents datasets and links outputs to the project's Zenodo community.

  IAM COMPACT aims to support the assessment of global climate goals, progress, and feasibility space, and to inform next-round NDCs beyond 2030 using an ensemble of IAMs and stakeholder engagement. Its maDMP (in ARGOS) states that datasets will be curated and linked to a Zenodo community, serving as the central, evolving plan for project data management. *(External context sources cited in text).*[IAM25, B+25]

- **Case Study 2 (Plant-Pollinator)** — a plan variant referencing the dataset *Plant-flower visitor interactions recorded in Argentina (Buenos Aires: Carlos Casares)* evaluated against the *WorldFAIR WP10: community-level FIP for plant-pollinator interactions data.* The dataset record summarizes two curated sub-datasets (522 and 2471 interaction records) with GBIF classification. [GVD24, WP124]

5.1.1 Mapping creation

Previously in Chapter 4 we discussed the implementation of the FIP questions harvester (take a nanopublication link to create a new mapping). To illustrate the produced output of this operation, Figure 5.1 shows the output for the FIP used in Case Study 1. Because of the size of the output, only the first three questions are included here to highlight the structure.


```json
{
  "FIP_Version": "v0.0.1",
  "FIP_maDMP_Mapping": [
    {
      "Question_URI": "https://w3id.org/fair/fip/terms/FIP-Question-F1-MD",
      "FAIR_principle": "F1",
      "FIP_question": "What globally unique, persistent, resolvable identifiers do you use for metadata records?",
      "DCS_field": "dataset.dataset_id.identifier",
      "Mapping_status": "Mapped",
      "Comments": "",
      "Allowed_values": [
        "DOI"
      ]
    },
    {
      "Question_URI": "https://w3id.org/fair/fip/terms/FIP-Question-F1-D",
      "FAIR_principle": "F1",
      "FIP_question": "What globally unique, persistent, resolvable identifiers do you use for datasets?",
      "DCS_field": "dataset.dataset_id.identifier",
      "Mapping_status": "Mapped",
      "Comments": "To be decided",
      "Allowed_values": []
    },
    {
      "Question_URI": "https://w3id.org/fair/fip/terms/FIP-Question-F2",
      "FAIR_principle": "F2",
      "FIP_question": "Which metadata schemas do you use for findability?",
      "DCS_field": "dataset.metadata.metadata_standard_id.identifier",
      "Mapping_status": "Mapped",
      "Comments": "",
      "Allowed_values": [
        "RO-Crate "
      ]
    },
```


Figure 5.1: Harvested questions from F2A Case Study 6: FAIR Climate Risk Assessments FIP. [B+25]

Execution. Each maDMP was evaluated against the selected mapping using both the CLI and the API. The system produced: an OSTrails FTR JSON-LD report, a compliance table (CSV/JSON), goal/validation summaries (JSON), and an TTL export for SPARQL (implemented for case study 2 to show its added value).

5.2 Mapping Coverage

Coverage indicates which fields from a FIP can be mapped to the schema of the DCS. Previously in Section 3.4.1, we showed which fields had a mapping, establishing that 5/21 fields could not be mapped. Thus, the maximum number of compliant fields can be 16. The evaluation is also dependent on whether allowed values are provided for each question: if no such values are available, the result is classified as *Indeterminate*, since no comparison can be made.

5.2.1 Coverage by case

The climate FIP provided allowed values for roughly half of the questions, so more test results (compliance) defaulted to *Indeterminate*. The plant-pollinator FIP offered allowed values for nearly all questions, permitting more determinate checks. This pattern highlights the practical influence of community-provided constraints on automation. Table 5.1 summarizes how many maDMP values and FIP allowed values (coverage) were provided by each case.

Table 5.1: Provided values per case study

Item	Climate goals	Plant-Pollinator
FIP: allowed values provided	11/21	20/21
maDMP values provided	9/21	11/21

5.3 Planned FAIR Alignment

5.3.1 Decision categories recalled

Per metric (one per FIP question), the evaluator outputs: *field status* Test Result (Present/Not Present) and a *compliance category* (Compliant / Non-compliant / Missing value / Not Applicable). These map to the canonical decision: *Pass* = Compliant; *Fail* = Non-compliant or Missing value; *Indeterminate* = Not Applicable.

5.3.2 Results by case

Table 5.2 shows the amount of values that were provided within each case study and the status of the compliance. Table 5.3 shows the overall decision/result for the test (result following the OSTrails FTR vocabulary for final prov:value).

Table 5.2: Observed counts per result category.

Case Study	Compliant	Non-compliant	Missing	Not Applicable
Climate goals	2/21	3/21	5/21	11/21
Plant-Pollinator	6/21	8/21	7/21	3/21

Table 5.3: Final decision per case (from *prov:value*) from Test Result.

Case study	Pass	Fail	Indeterminate
Climate	2/21	8/21	11/21
Plant-Pollinator	6/21	12/21	3/21

We can interpret this as follows: when clear allowed values and explicit DCS fields are available (for example, identifiers or licenses), results tend to be marked as *Compliant*. In contrast, aspects where community practice is more detailed than what the DCS captures (such as authentication details in A1.2) often lead to *Partially Mapped* or *Not Applicable* outcomes. The *climate goal* case study shows more *Indeterminate* results because fewer FIP questions specify allowed values. *The plant-pollinator* case study, on the other hand, enables more definite checks but also produces more *Fail* results when plan entries are missing or conflict with community expectations.

5.3.3 OSTRAILS compliant output

To demonstrate how results are formatted, Figure 5.2 shows an excerpt of the OSTrails-compliant JSON-LD output produced for one FIP question, including a Metric, Test and Test Result.

5.4 Early-Stage Plans evaluation

To evaluate how much the evaluator can say before a dataset exists, we created a controlled variant of the plant–pollinator plan: we duplicated the original maDMP, removed only the dataset reference (PID/URL), and left all other fields unchanged (e.g., intended license, target repository, metadata standard). We then re-evaluated this early-stage variant against the same community FIP. Results can be seen in Tables 5.4 and 5.5

Table 5.4: Observed counts per result category for the plant-pollinator (without dataset).

Case Study	Compliant	Non-compliant	Missing	Not Applicable
Plant-Pollinator (no dataset link)	4/21	7/21	7/21	3/21

This experiment shows that even without a dataset link, the evaluator still judged most metrics because many checks depend on planned choices (identifier scheme, license, repository/hosting, metadata standard, access statements) rather than on a concrete

```json
{
  "@id": "#FIP19.Q19",
  "@type": "dqv:Metric",
  "dcterms:identifier": "FIP19.Q19",
  "dcterms:title": "Which usage license do you use for your datasets?",
  "dcterms:description": "Which usage license do you use for your datasets?",
  "dcat:version": ""
},
{
  "@id": "#Test_FIP19.Q19",
  "@type": [
    "ftr:Test",
    "dcat:DataService",
    "prov:Agent"
  ],
  "dcterms:identifier": "Test_FIP19.Q19",
  "dcterms:title": "Which usage license do you use for your datasets?",
  "dcterms:description": "DCS field: dataset.distribution.license.license_ref; allowed_value from the community: [\"CC BY 4.0\", \"CC0 1.0\", \"CC BY-NC 4.0 \"]",
  "dcterms:license": "https://creativecommons.org/publicdomain/zero/1.0/",
  "dcat:version": "",
  "sio:is-implementation-of": "#FIP19.Q19",
  "ftr:testMetric": "#FIP19.Q19"
},
{
  "@id": "#Test_FIP19.Q19_result",
  "@type": [
    "ftr:TestResult",
    "prov:Entity"
  ],
  "dcterms:identifier": "Test_FIP19.Q19",
  "dcterms:title": "Result for FIP19.Q19",
  "dcterms:description": "Field status: Present; maDMP value: [\"https://creativecommons.org/licenses/by/4.0/\"]; compliance: Compliant",
  "dcterms:license": "https://creativecommons.org/publicdomain/zero/1.0/",
  "prov:value": [
    "pass"
  ],
  "ftr:log": [
    "https://creativecommons.org/licenses/by/4.0/"
  ],
  "ftr:completion": "100",
  "ftr:outputFromTest": "#Test_FIP19.Q19",
  "prov:wasDerivedFrom": "#input_dmp"
},
```


Figure 5.2: Excerpt of OSTrails FTR in JSON-LD format for one FIP question.

Table 5.5: Final decision for the plant-pollinator (without dataset) (from *prov:value*).

Case study	Pass	Fail	Indeterminate
Plant-Pollinator (no dataset link)	4/21	14/21	3/21

dataset. Compared to the original maDMP, the early-stage variant showed a lower number of *Compliant/Pass* outcomes and a higher number of *Non-compliant/Fail* outcomes, while *Indeterminate* remained the same with the idea that removing resolvable dataset affects some metrics/tests, but leaves other planned-choice checks intact.

5.5 SPARQL Explorations over FTR

We used the JSON-LD to TTL file converter on the maDMP used for case study 2 (Section 5.1) to show how SPARQL queries can help us summarize outcomes and extract evidence(see Figure 5.3). An example can be seen in Figures 5.4.

5.6 Comparison and Context

In comparison with manual reviews, this tool replicates (in an automated way) presence, allowed-value membership, and cross-field consistency, returning explainable evidence and machine-actionable outputs. Human judgment remains useful for nuanced infrastructure choices (often partially mapped). This prototype adds FIP-driven planning-time *guideline*

```python
from rdflib import Graph

g = Graph()
g.parse(r"../results/Plant-flower_visitor_interactions_ostrails_results.ttl", format="ttl")

print(f"Graph has {len(g)} triples.")
```

```
✓ 0.1s
Graph has 734 triples.
```

Figure 5.3: Setup to prepare RDF graph for semantic querying.

```python
# Metrics and name of benchmark for F1
query = """
PREFIX ftr: <https://w3id.org/ftr#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?metric ?title
WHERE {
    ?benchmark ftr:hasAssociatedMetric ?metric ;
               dcterms:identifier "F1" ;
               dcterms:title ?title .
}
"""
for row in g.query(query):
    print(row.metric, row.title)
```

```
✓ 0.0s
#FIP01.Q1 Identifier type
#FIP02.Q2 Identifier type
```

Figure 5.4: Print metrics belonging to Benchmark *Identifier type*.

compliance and standardized outputs (OSTrails FTR) that support downstream semantic analyses..

Dataset-level evaluation is out of scope for this thesis. However, to illustrate the *quality of actions* perspective and to contrast existing dataset FAIR tools with our *compliance* evaluator, we include an example of a F-UJI run, shown in Figure 5.5. F-UJI summarizes FAIRness of the published object (e.g., PID presence/resolvability, metadata sufficiency, license detection, protocol use).

5.7 Summary and Discussion

This chapter showed the prototype running on two real cases and one created early–stage variant. The results confirmed that the evaluator can process maDMPs, evaluate against domain-specific community guidelines (FIP), and return reproducible outputs in the OSTrails FTR format, together with human–readable outputs. The climate goal case

Figure 5.5: Illustrative F-UJI dataset-level FAIRness evaluation for dataset with DOI `https://zenodo.org/record/6467615`. [DHF25]

study demonstrated that when fewer allowed values are provided in the FIP, many results remain indeterminate, while the plant–pollinator case study showed how a nearly complete set of allowed values enables more decisive checks but also exposes gaps in the maDMP more clearly. The controlled case, where the dataset link was removed, confirmed that the tool remains useful even before a dataset exists. Most questions can still be evaluated based on planned choices such as the intended license, repository, or metadata standard, and the absence of a dataset reference mainly affects findability and access–related items. This supports the idea that the evaluator can guide researchers already at an early planning stage.

The experiments also underline some practical lessons. Questions with well–established catalogs and explicit DCS fields, such as identifiers or licenses, consistently produced compliant results when filled, while aspects where community practice is more detailed than the DCS (for example authentication in A1.2) tended to produce partially mapped or not applicable outcomes. The distinction between "fail" and "indeterminate" proved important: a fail result pointed either to a real misalignment or a missing value in the plan, whereas indeterminate highlighted that the community had not provided allowed values or that no clear mapping exists. Rather than guessing, the tool makes these limitations explicit, which can help both data stewards and researchers to understand where improvements are possible.

Beyond the numbers, this evaluation shows the added value of making outputs interoperable. By exporting results in JSON-LD and converting them to RDF representations, it was possible to run simple SPARQL queries to summarise results or extract evidence, something that could support dashboards or comparative studies across many plans. This approach opens the door to automated monitoring of compliance with community FIPs. At the same time, it is clear that dataset–level FAIRness checks remain out of scope for the evaluator itself. Tools like F-UJI can complement this work once a dataset is available, as discussed in the next chapter.

Overall, the prototype demonstrates that automated guideline compliance evaluation of maDMPs is feasible and already useful in practice. It provides clear signals about what can be checked now and what remains unspecified, and allows quick iterations: a researcher can update their plan, re–run the tool, and immediately see how the status changes. The full implementation is available at `https://github.com/bernardobap21/`

DMP-Evaluation-through-integration-of-maDMPs-and-FIPs, where the repository can be cloned. In addition, a running instance of the API can be accessed at https://dmp-evaluation.onrender.com/docs, which exposes the main functionality via simple endpoints. Together these resources make it possible to reproduce the results in this chapter and to experiment with other maDMPs and FIPs.

CHAPTER 6

Conclusion and Future Work

This thesis presented the automated evaluation of maDMPs against domain-specific community expectations captured in FIPs. The core contribution is an operational standards-driven evaluator that loads a selected FIP and its allowed values (or creates it from a Nanopublication), applies mapping to DCS fields, and emits evidence-bearing decisions per FIP question (treated as metrics) following the OSTrails FAIR Test Results (FTR) vocabulary, with an optional conversion to TTL for semantic exploration. The design emphasizes transparency: each result carries the observed value, the checked DCS path, and the constraint reference, so reviewers see precisely *what* was judged and *why*.

Beyond guideline compliance, the evaluator also outputs reports for completeness and feasibility (accuracy/availability/consistency) goals as both human and machine-readable summaries, but the main focus throughout is the FIP-driven compliance check during different phases of the DMP. The implementation offers both CLI and API interfaces and supports harvesting new profiles from nanopublications, making it practical to evaluate maDMPs against different domains. Two case studies (climate risk assessment and plant-pollinator interactions) and a controlled early-stage variant showed the method in practice.

6.1 Addressing the Research Questions

In this section, we discuss the insights that our research questions presented at the beginning of this thesis produced.

Research question 1. To what extent can machine actionable Data Management Plans (maDMPs) and FAIR Implementation Profiles (FIPs) be aligned to enable automated evaluation of Data Management Plans?

We produced a FIPs to DCS mapping catalogue covering all 21 FIP questions (modeled as metrics), classifying each alignment as Mapped, Partially Mapped, or Not Mapped.

This catalogue defines the scope for automation: questions with a reliable DCS path and community allowed values are machine-checkable; partially mapped items reflect granularity gaps (e.g., A1.2 authentication/authorization details), and unmapped items are out of scope for now. In short, RQ1 is answered by a concrete, reusable mapping and a measurement of mapping coverage that makes the scope and limits of automation explicit. [TP23, Arn24, Res21, SM+20]

Research question 1.1. Which of the DMP evaluation goals can be supported by FIPs, and to what extent can these be automated? FIPs are by design machine-actionable. They make automated guideline checks possible because they define the community-defined allowed values for each question. By comparing the values in a maDMP against these constraints, the evaluator can automatically determine whether a plan follows the expected guidelines. Other evaluation goals—such as checking completeness (is the field present?) and feasibility (accuracy, availability, consistency) can be addressed directly through the DCS schema and extra validation logic, but these do not rely on FIPs. [TP23, Arn24]

Research question 2. To what extent can the intended actions in maDMPs be evaluated against FAIR-enabling expectations defined in community-specific FIPs? Particularly when data has not yet been generated or lacks repository identifiers. Within the scope of RQ1, the prototype automatically evaluates planned actions: given a DCS-compliant JSON plan and a selected FIP with allowed values, the evaluator returns a canonical decision (Pass, Fail or Indeterminate) along with the field status (Present/Not Present) and a compliance category (Compliant / Non-compliant / Missing value / Not Applicable). This approach is useful even before datasets exist, since users can already fill in certain fields and immediately check whether those values align with community expectations. In addition, the prototype provides a clear compliance table and the mapping used, so researchers can quickly see what their community expects—without needing to read through extensive documentation.

The case studies in Chapter 5 illustrate two boundary conditions that explain the results: where the FIP does provide allowed values for most questions (plant–pollinator: 20/21), more checks are determinate, producing clear Pass or Fail outcomes and actionable recommendations and when the FIP provides fewer allowed values (climate risk: 11/21), more results are Indeterminate by design, signaling that a community decision is missing (domain-specific decisions have a big impact in the evaluation). A controlled early-stage probe (removing only the dataset link) showed that useful evaluation still happens on the planned-choice metrics; the absence of a PID/URL mainly reduces findability/access signals and increases Fail or Missing in those areas.

6.2 Contributions

This thesis makes several contributions to the ongoing discussion about how to make DMPs more machine-actionable and assessable against community guidelines. First, it introduces a community-guided design for guideline compliance: a conceptual framework that operationalizes FIPs over DCS maDMPs and exports results as OSTrails FTR, ensuring reproducible, provenance-rich, and semantically queryable assessments. Building on this design, the work develops a mapping catalogue, where each of the 21 FIP questions is aligned to a DCS field, thus making the scope and limits of automation explicit and measurable.

Another contribution is the development of an executable evaluator with explainable outputs. In this system, each FIP question corresponds to one metric and is tested to produce a result carrying a canonical decision (*prov:value*) along with human-readable evidence, while optional TTL serialization enables downstream SPARQL exploration. Furthermore, the work implements Nanopublication ingestion and stable ordering. Through a dedicated harvester and a Question Map, FIP question URIs are bound to FAIR sub-principles, canonical text, and DCS selectors, thereby enforcing a stable FAIR order in reports and improving readability [SMK+22, SM+20].

Finally, the thesis contributes case-study evidence for compliance use. By evaluating two DMPs and an early-stage variant, it demonstrates that guideline compliance can be assessed even before datasets exist. In this way, missing or non-compliant fields become visible as Fail or Missing outcomes that researchers can correct themselves while Indeterminate results highlight areas where the community needs to publish clearer guidance. Together, these contributions show how community guidelines represented as FIPs can be translated into actionable, evidence-bearing checks that improve the quality and transparency of maDMPs.

6.3 Limitations

While this thesis produced encouraging and meaningful results, certain limitations must be acknowledged to provide a balanced view of the prototype's capabilities:

One limitation of the current approach is its dependence on community guidelines. If a FIP does not provide allowed values for a particular question, the evaluator cannot make a judgment. In these cases, the result is marked as *Indeterminate* or *Not Applicable.* This means that the quality of the automated assessment is tied to how complete and detailed the community guidance is.

Another limitation is the issue of *granularity mismatch.* Some community expectations are described at a much more detailed level than the information that can be expressed in the DCS. For example, authentication and authorization mechanisms (A1.2: Which authentication & authorisation technique do you use for metadata records?) may require specific details, while the DCS only captures broader categories (Open, Shared, Close). This leads to partially mapped questions, where the tool can give only limited or conservative results.

There is also the problem of domain-specific catalogs. For certain areas, such as licenses, well-maintained registries like SPDX exist and can be used directly. However, for many other fields there are no canonical lists, and the evaluator must fall back on smaller curated lists or regular expressions. These methods are useful but incomplete, and they may miss community practices that are not yet standardized.

A further limitation is the scope of the prototype. This thesis addresses goals G1, G2 and G4 (see Subection 2.8.1), focusing on guideline compliance checks of maDMPs against FIPs. It does not evaluate the FAIRness of actual datasets. Such dataset-level assessment could be added later by connecting to external tools like F-UJI, but it is not part of the current implementation.

Finally, there are some technical constraints to note. If a FIP is not available in nanopublication form, the evaluator cannot automatically extract the questions and allowed values needed for the mapping. At present, the tool cannot run multiple evaluations in parallel, which limits scalability when working with larger collections of plans. There is also a dependency on external standards: if the structure of the DCS schema or the OSTrails FAIR Test Results specification changes in the future, the implementation would need to be updated to remain compatible.

6.4 Implications for Stakeholders

For researchers and data stewards, the evaluator shows that it is possible to check guideline compliance already during the planning phase of a project. Running these checks early allows them to identify *Fail* or *Missing* items and fix them quickly, for example by adding the intended license, the planned repository, or the identifier scheme. The results marked as *Indeterminate* should be seen as useful signals: they point to areas where either the plan needs more detail or where the chosen FIP does not yet provide clear enough guidance.

For funders and institutional reviewers, the standardized FTR outputs make it possible to examine submissions more efficiently. Where guideline compliance can be checked automatically, results provide immediate evidence; where outcomes are indeterminate, they highlight gaps that communities need to address in order to make evaluation more robust and consistent. In this way, reviewers can focus attention on the parts of a plan that truly require human judgment.

For community maintainers and authors of FIPs, the findings emphasize the importance of publishing allowed values and keeping them up to date. Doing so directly increases the proportion of automatable checks and reduces the number of *Indeterminate* results. Using nanopublications ensures that provenance is clear and that different versions of a profile can be tracked and reused reliably.

6.5 Future Work

Several directions for future work emerged during this thesis. They range from technical extensions of the prototype to broader community and workflow integration.

6.5.1 External tools to cover goal quality of actions

One of the most natural extensions is to connect the evaluator with dataset-level FAIRness checks. An optional integration with the F-UJI API could be added so that, whenever a dataset PID or URL is available, the evaluator presents both views side by side: the planned intent from the maDMP and FIP, and the realized practice from the dataset itself. This would not replace the planning-time compliance focus of the current work but rather complement it, giving reviewers and researchers a complete picture [DH21, DHF25].

6.5.2 Improving mapping coverage and tooling

Another important area is the expansion and refinement of the mapping catalogue. Semi-automatic mapping suggestions could be generated by comparing FIP question text with DCS fields or by mining patterns across existing plans. These suggestions could then be curated through interactive interfaces designed for data stewards. Building cross-domain mapping libraries would also help communities reuse work and avoid duplication.

6.5.3 Managing constraint catalogues more robustly

Constraint catalogues—such as lists of identifier schemes, repositories, or vocabularies should be created, published in an open format and used instead of regular expressions since they are easier to manage and update. Like it was implemented for SPDX license. This would increase trust in the evaluator's results and make runs more reproducible.

6.5.4 Adding version control and run storage

At present, results can be generated and downloaded locally, but they are not stored or versioned online. A useful extension would be to integrate version control for evaluations, so that each run is recorded with its mapping, catalog versions, software version, and timestamp. Storing runs in a shared repository or provenance store would make it possible to trace how results were obtained, compare different evaluations over time, and reproduce earlier validations. This would strengthen trust in the outputs and provide a clearer audit trail for both researchers and reviewers.

6.5.5 Integration into researcher workflows

Finally, to maximize its impact, the evaluator should be embedded in the tools that researchers already use. Integrating it into platforms such as ARGOS or the DSW, and into institutional dashboards, would allow researchers to run "edit → re-run" cycles as part of their everyday proposal and project workflows. This would normalize the use of guideline compliance checks and reduce the burden on both authors and reviewers [TP23, OST25c, OST25a, OST25b].

In conclusion, this work demonstrates that automated, evidence-bearing assessment of planned actions in maDMPs is both feasible and useful when grounded in FIPs. The approach is intentionally simple and transparent: a clear mapping, conservative rules, and standardized FTR results that travel well to dashboards and knowledge graphs. While dataset-level FAIRness remains a separate layer, the two perspectives are complementary: alignment checks help researchers before data release; dataset-level tools verify after. Together, they point toward a more transparent ecosystem for research data management.

Overview of Generative AI Tools Used

In this thesis, the following tools were used as support:

- **ChatGPT-4o and 5** mainly for proofreading, checking the structure of paragraphs for example using the prompt: *"Is this text clear enough given that I want to express this: [input text]"* or *"Is the grammar correct and appropriate for academic purposes"*. This tool was only used as writing aid, always being manually checked to preserve the original intention and style of writing.
- **DeepL** to help translate certain ideas to English and German. For example for the Kurzfassung.

Übersicht verwendeter Hilfsmittel

- **ChatGPT-4o und 5** wurden hauptsächlich für das Korrekturlesen sowie zur Überprüfung der Absatzstruktur verwendet, z. B. mit den Prompts: *Ïs this text clear enough given that I want to express this: [input text]"* oder *Ïs the grammar correct and appropriate for academic purposes"*. Dieses Tool wurde ausschließlich als Schreibhilfe eingesetzt und die Ergebnisse wurden stets manuell überprüft, um die ursprüngliche Intention und den Schreibstil zu bewahren.
- **DeepL** wurde genutzt, um bestimmte Ideen ins Englische und Deutsche zu übersetzen, beispielsweise für die Kurzfassung.

List of Figures

List of Tables

Glossary

benchmark A standard or point of reference against which things may be compared, evaluated, or measured, often used to assess performance or compliance. 14

completeness The extent to which a data management plan covers all relevant aspects, sections, and questions necessary to address the requirements of a given template, guideline, or FAIR Implementation Profile. 3, 18, 21, 40

dot-path A notation for accessing nested elements in structured data (e.g., JSON) using a sequence of keys separated by dots. For example, `dataset.distribution.license.license_ref` refers to the `license_ref` field nested inside the distribution of a dataset. Dot-path access is often implemented in a *safe* manner, meaning that missing fields return a neutral value instead of causing an error.. 39, 42–45, 47

FAIR assessment tool A software service or framework that evaluates the degree to which digital objects or data management practices comply with the fair principles, often by running automated checks on identifiers, metadata, accessibility protocols, vocabularies, and licensing (e.g., F-UJI, FAIR evaluator, maturity models). 16

FAIR-enabling resource A standard, service, tool, or vocabulary selected by a community to implement a specific FAIR sub-principle (e.g., DOI service, metadata schema, repository, license). 11

feasibility The practicality of implementing the measures, actions, and resources described in a data management plan within the project's constraints, such as budget, skills, and infrastructure. 3, 18, 21, 40

FIP ontology A formal model that represents FAIR Implementation Profiles, their declarations, FAIR sub-principles, and FAIR-enabling resources, enabling machine-actionable profiles. 11

GO FAIR An international initiative that promotes the practical implementation of the FAIR principles through community-driven Implementation Networks and the development of shared practices and resources. 5, 11, 16

guideline compliance The degree to which the content of a data management plan adheres to the recommendations, standards, or community guidelines specified by a particular policy or framework. 3, 21, 40

HEAD request An HTTP request method used to retrieve only the headers of a resource, without downloading its body. Often employed to check the availability, validity, or metadata of a URL (e.g., for testing whether a persistent identifier resolves correctly).. 45, 48

machine-actionable Refers to information that is structured in a way that enables software agents to automatically interpret, process, and act upon it without requiring manual intervention. In the context of Data Management Plans, this allows validation, transformation, and integration with other systems.. 7, 19, 22, 60

mapping coverage The proportion of FIP questions that can be mapped to corresponding fields in the DCS schema. xi, xiii, 3, 4

Nanopublication A structured, granular publication format representing assertions as RDF triples, each with provenance, enabling precise citations and machine readability.. xi, xiii, 11, 22, 29, 37, 39, 40, 59, 61

NDC Nationally Determined Contributions; country-level climate action plans submitted under the Paris Agreement, outlining targets and measures to reduce greenhouse gas emissions and adapt to climate change. 51

planned FAIR alignment The degree to which the intended actions described in a maDMP align with FAIR-enabling expectations defined in a FAIR Implementation Profile. xi, xiii, 3, 4

PROV The W3C Provenance Ontology for representing provenance information. 14

proxy indicator An indirect measure used when direct measurement of a variable is not possible; it should correlate with the target variable. 4

regular expression A sequence of characters that defines a search pattern, typically used for string matching, validation, and text processing.. 39, 48, 63

Semantic Web An extension of the World Wide Web that enables data to be shared, linked, and processed by machines using formal knowledge representation languages such as RDF, OWL, and SPARQL. 5

SPDX licenses A standardized list of open source and other software/data licenses maintained by the Software Package Data Exchange (SPDX) project of the Linux Foundation. SPDX provides unique identifiers for licenses (e.g., “CC-BY-4.0”) to ensure unambiguous reference and automated compliance checking. 5, 39, 43, 44, 48, 49

Acronyms

API Application Programming Interface. 2, 14, 16, 21, 37, 39, 40, 49, 53

CLI Command-Line Interface. 21, 37, 39, 40, 46, 49, 53

DCAT Data Catalog Vocabulary. 11, 14

DCS Data Management Plan Common Standard. xi, xiii, xvi, 1–5, 7, 14–19, 21–25, 27–30, 37, 39–42, 44, 45, 47, 53, 54, 57, 59–62

DCSO DMP Common Standard Ontology. 16, 27

DMP Data Management Plan. 1–3, 7, 8, 14, 16, 17, 19, 21, 22, 37, 51, 59–61, 69

DMP-IF DMP Interoperability Framework. 14

DOI Digital Object Identifier. 11, 14, 16, 57, 69

DQV Data Quality Vocabulary. 14

DSR Design Science Research. 4

DSRM Design Science Research Methodology. 4

DSW Data Stewardship Wizard. 11, 63

F-UJI FAIRsFAIR Automated FAIR Data Assessment Tool. 2, 5, 15, 16, 27, 51, 56, 57, 62, 63

FAIR Findable, Accessible, Interoperable, and Reusable. 1–3, 5, 8, 11, 16, 18, 51, 56

FAIR-IF FAIR Interoperability Framework. 14, 19

FER FAIR-enabling resource. 11, 51

FIP FAIR Implementation Profile. xi, xiii, xvi, 1–5, 11, 14–16, 19, 21–25, 27–32, 37, 39–41, 45, 47–49, 51–63, 69

FIP2DMP FAIR Implementation Profile to Data Management Plan. 14

FR Functional Requirements. 24

FTR FAIR Test Results (OSTrails model). xi, xiii, xvi, 14, 15, 32, 35, 39, 40, 45, 46, 48, 51, 53, 55, 56, 59, 61, 62, 64, 69

IAM Integrated Assessment Model. 52

JSON JavaScript Object Notation. 1, 27, 44, 47, 60

JSON-LD JavaScript Object Notation for Linked Data. 11, 14, 15, 19, 21, 29, 40, 45, 47–49, 53–55, 57, 69

maDMP Machine-actionable Data Management Plan. xi, xiii, 1–3, 5, 7–9, 14–19, 21–23, 25, 27–30, 32, 37, 39, 42, 44, 46–49, 51, 53–61, 63, 69

NFR Non-Functional Requirements. 24

OSTrails Open Science Training and Resources for Advanced Interoperable Learning Systems. xi, xiii, xvi, 2, 4, 5, 14, 19, 21, 23–25, 27, 32, 35, 37, 39, 40, 45, 47–49, 51, 53–56, 59, 61, 62, 69

PID Persistent Identifier. 16

RDA Research Data Alliance. 1, 7

RDF Resource Description Framework. xi, xiii, 4, 11, 15, 19, 25, 27, 47, 48, 57

RDM Research Data Management. xi, xiii, 1, 7

SKG Scientific Knowledge Graphs. 14, 16, 17

SPARQL SPARQL Protocol and RDF Query Language. 4, 16, 19, 23, 25, 27, 40, 48, 51, 53, 55, 57, 61

TTL Terse RDF Triple Language. 2, 21, 29, 40, 47–49, 53, 55, 59, 61

URI Uniform Resource Identifier — a standardized string of characters used to identify a resource either by location, name, or both. It generalizes URLs and URNs.. 39, 40, 42

Bibliography

[Arn24] Lukas Arnhold. Automated quality indicators for machine-actionable data management plans. Master's thesis, TU Wien, 2024.

[B+25] Julia Bartsch et al. F2a case study 6: Fair climate risk assessments. FAIR nanopublication, 2025.

[CCE+22] João Cardoso, Leyla Jael Castro, Fajar J. Ekaputra, Marie C. Jacquemot, Marek Suchánek, Tomasz Miksa, and José Borbinha. Dcso: Towards an ontology for machine-actionable data management plans. *Journal of Biomedical Semantics*, 13(1):21, 2022.

[con24] Wikipedia contributors. Proxy (statistics), 2024.

[con25] Wikipedia contributors. Design science (methodology), 2025.

[CWL14] Richard Cyganiak, David Wood, and Markus Lanthaler. Rdf 1.1 concepts and abstract syntax. `https://www.w3.org/TR/rdf11-concepts/`, 2014. W3C Recommendation.

[DH21] Anusuriya Devaraju and Robert Huber. An automated solution for measuring the progress toward fair research data. *Patterns*, 2(11):100370, 2021.

[DHF25] Anusuriya Devaraju, Robert Huber, and FAIRsFAIR. F-uji: Automated fair data assessment tool, 2025.

[FAI24] FAIR-IMPACT. FIP Wizard catalogue entry, 2024. "FIPs are published by the FIP Wizard as FAIR Open Data (nanopublications).".

[GO 24] GO FAIR Foundation. FIP Wizard User Guide, 2024. "FIP Wizard can consume and produce nanopublications.".

[GO nd] GO FAIR. Fair implementation profile. `https://www.go-fair.org/how-to-go-fair/fair-implementation-profile/`, n.d.

[GO25] Daniel Garijo and OSTrails Team. Fair test results (ftr) specification. `https://w3id.org/ftr`, 2025. See also repository: `https://github.com/OSTrails/FAIR_assessment_output_specification`.

[GVD24] Rocío Ana González-Vaquero and Mariano Devoto. Plant-flower visitor interactions recorded in argentina (buenos aires: Carlos casares county). Zenodo, 2024.

[GW+25] Daniel Garijo, Mark D Wilkinson, et al. OSTrails FAIR Assessment Output Specification v1.1.0, 2025.

[HMG+23] Kristina Maria Hettne, Barbara Magagna, Alessa An Gambardella, Marek Suchánek, Fieke Schoots, and Erik Schultes. Fip2dmp: Linking data management plans with fair implementation profiles. *FAIR Connect*, 1(1):23–27, 2023.

[HMPR04] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.

[HS13] Steve Harris and Andy Seaborne. Sparql 1.1 query language. `https://www.w3.org/TR/sparql11-query/`, 2013. W3C Recommendation.

[IAM25] IAM COMPACT project. Iam compact – machine-actionable data management plan (new). Zenodo, 2025. Output management plan; includes JSON and PDF versions.

[Lin23] Linux Foundation SPDX Workgroup. Spdx: Software package data exchange. `https://spdx.org/licenses/`, 2023.

[Mon17] B. et al. Mons. Cloudy, increasingly fair; revisiting the fair data guiding principles for the european open science cloud. *Information Services Use*, 37(2):49–56, 2017.

[MSK25] Barbara Magagna, Erik Schultes, and Tobias Kuhn. Fair implementation profile (fip) ontology. `https://peta-pico.github.io/FAIR-nanopubs/fip/index-en.html`, 2025.

[MWN19] Tomasz Miksa, Paul Walk, and Peter Neish. Ten principles for machine-actionable data management plans. *PLOS Computational Biology*, 15(3):e1006750, 2019.

[MWN+21] Tomasz Miksa, Paul Walk, Peter Neish, Simon Oblasser, Hollydawn Murray, Tom Renner, Marie-Christine Jacquemot-Perbal, João Cardoso, Trond Kvamme, Maria Praetzellis, Marek Suchánek, Rob Hooft, Benjamin Faure, Hanne Moa, Adil Hasan, and Sarah Jones. Application profile for machine-actionable data management plans. *Data Science Journal*, 20:1–24, 2021.

[OST25a] OSTrails Consortium. Architecture: Dmp-if (data management plan interoperability framework), 2025.

[OST25b] OSTrails Consortium. Architecture: Fair-if (fair testing interoperability framework), 2025.

[OST25c] OSTrails Consortium. Ostrails reference architecture v1: The foundation for interoperable research tools, 2025.

[PTG+20] Ken Peffers, Tuure Tuunanen, Charles E. Gengler, et al. Design science research process: A model for producing and presenting information systems research, 2020.

[RDA25] RDA-DMP-Common Standard Working Group. maDMP-schema-1.2.json: JSON Schema for maDMP version 1.2. `https://github.com/RDA-DMP-Common/RDA-DMP-Common-Standard/blob/master/examples/JSON/JSON-schema/1.2/maDMP-schema-1.2.json`, 2025.

[Res21] Research Data Alliance. RDA DMP Common Standard, 2021.

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

[SKL20] Manu Sporny, Gregg Kellogg, and Dave Longley. Json-ld 1.1. `https://www.w3.org/TR/json-ld11/`, 2020. W3C Recommendation.

[SM+20] Erik Schultes, Barbara Magagna, et al. Reusable FAIR Implementation Profiles as accelerators for FAIR convergence. *Data Intelligence*, 2(1-2):186–200, 2020.

[SMK+22] Erik Schultes, Barbara Magagna, Tobias Kuhn, Marek Suchánek, Luiz Olavo Bonino da Silva Santos, and Barend Mons. The comparative anatomy of nanopublications and fair digital objects. *Research Ideas and Outcomes (RIO)*, 8:e94150, 2022.

[TP23] Miksa Tomasz M. Slifka J. Knaisl V. Ekaputra F.J. Kovacevic F. Ningtyas A.M. El-Ebshihy A. T., Suchánek and R. Pergl. Towards a toolbox for automated assessment of machine-actionable data management plans. *Data Science Journal*, 21:28, 2023.

[WDA+16] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, et al. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016.

[WDS+19] Mark D. Wilkinson, Michel Dumontier, Susanna-Assunta Sansone, Luiz Olavo Bonino da Silva Santos, Mariano Prieto, Domenico Batista, Peter McQuilton, Tobias Kuhn, Philippe Rocca-Serra, Mercè Crosas, and Erik Schultes. Evaluating FAIR maturity through a scalable, automated, community-governed framework. *Scientific Data*, 6(1):174, 2019.

[WP124] WorldFAIR WP10. Worldfair wp10: community-level fip for plant-pollinator interactions data. Zenodo, 2024.