Evaluation of Architectures for FAIR Data Management in a Research Data Management Use Case

Benedikt Heinrichs[a], Marius Politze[b] and M. Amin Yazdi[c]
IT Center, RWTH Aachen University, Seffenter Weg 23, Aachen, Germany

Keywords: Research Data Management, FAIR Data Management, Digital Objects, Metadata, FAIR Data Architectures.

Abstract: Research data management systems are mostly designed to manage data according to the FAIR Guiding Principles. In order for the systems themselves to follow this promise and improve the possibility of networking between decentralized systems, they should incorporate standardized interfaces for exchange of data and metadata. For this purpose, in the last couple of years, several standards emerged which try to fill this gap and define data structures and APIs. This paper aims to evaluate these standards by defining the requirements of a research data management system called Coscine as a use case and seeing if the current standards meet the defined needs. The evaluation shows that there is not one complete standard for every requirement but that they can complete each other to fulfill the goal of a standardized research data management system.

1 INTRODUCTION

The FAIR Guiding Principles (Wilkinson et al., 2016) are a critical part of today's research environment. Making research data and their metadata findable, accessible, interoperable, and re-usable as FAIR data or in the form of FAIR digital objects therefore has a central place in research data management systems and initiatives like NFDI4Ing (Schmitt et al., 2020). The field of research data management tries to bring the integration of FAIR principles for research data together, however solutions for how to implement these principles diverge (Jacobsen et al., 2020) and creating an overview is not always the easiest task. Therefore, research data management systems like Coscine (Politze et al., 2020) sometimes build their own interfaces instead of following defined standards on interacting with FAIR data. This, of course, leads to even more divergence and prevents the aim of following the FAIR Guiding Principles. For this reason, this paper aims to provide an overview on open standards, which claim to provide data structures and APIs for building an architecture to enhance FAIR data management. Based on the described research data management system, Coscine, requirements for (meta-)data management will be collected. Based on them, the standards are evaluated, and a description is given on how they fit into each requirement. Especially, important points like data provenance, persistent identification and metadata management in the linked data environment will be considered. Additionally, based on the use case and its requirements, it will be evaluated how well these standards fit into a real-world example and at which level they could be applied and integrated. Finally, the implementation decision for the real-world example will be discussed and described.

1.1 Use Case

By working with researchers across different domains and the advent of the FAIR Guiding Principles, a need was made clear: A platform for facilitating research data management and metadata annotation. However, a specific need from the researchers was that their own research data could be located at separate storage system providers, so a platform would need to account for these different locations. Most storage systems are coming from the commercial area and therefore do not adhere to the FAIR principles. To overcome this challenge, an intermediate layer was needed that can make arbitrary storage systems "FAIR". With these requirements, Coscine (Politze et al., 2020) was born and developed at the RWTH Aachen University by the research data management team. It is advertised as a (C)ollaborative (Sc)ientific (In)tegration

[a] https://orcid.org/0000-0003-3309-5985
[b] https://orcid.org/0000-0003-3175-0659
[c] https://orcid.org/0000-0002-0628-4644

Heinrichs, B., Politze, M. and Yazdi, M.
Evaluation of Architectures for FAIR Data Management in a Research Data Management Use Case.
DOI: 10.5220/0011302700003269
In *Proceedings of the 11th International Conference on Data Science, Technology and Applications (DATA 2022)*, pages 476-483
ISBN: 978-989-758-583-8; ISSN: 2184-285X

(E)nvironment which aims to support the researchers during their active research phases, where research data artifacts are generated and changed across a variety of different storage providers. To organize the different research activities of scientists, Coscine supports the creation of projects. These projects can be shared across other scientists for collaboration possibilities and can have an arbitrary number of storage resources. These resources are assigned a globally unique persistent identifier within the platform to make them easily shareable. The uploaded research data has to be annotated by metadata which is based on semantic web technologies like RDF (Wood et al., 2014) and validated by SHACL (Kontokostas and Knublauch, 2017).

In the first approach, Coscine was built according to the scientists' requirements. As the system now strives for implementing a more standardized approach, it needs to be assessed how common standardized FAIR data interfaces can represent the required complexity.

1.2 Research Goal

Following up on the introduced area and the use case, the goal of this paper is to answer the following research question: "To what extent are current FAIR data interfaces suitable to service and exchange research data and metadata in a decentralized environment?". The answer to this question will lead to a concrete idea on how to proceed with the inclusion of standards for the use case.

2 EXISTING ARCHITECTURAL STANDARDS

For the evaluation part of this paper, a short overview of the research area and the architectural standards is given in this section. The separate standards are evaluated in more detail in the later sections.

2.1 Persistent Identifiers (PIDs)

The persistent resolution of a web resource is an essential task, making standard URLs not an ideal solution, since external factors like the change of the web-route or a company going out of business could mean that the resolution is not possible anymore. This issue is something persistent identifiers (PIDs) claim to fix by having an identifier with an updatable pointer. A system like the Handle system (Sun et al., 2003) provides a clear registration and resolver system for these PIDs, which can provide the needed reference to a web resource over time. Solutions like ePIC (Schwardmann, 2015) go a step further, provide clear implementations and encourages PID information records containing metadata like information types.

2.2 Digital Object (DO)

The concept of Digital Objects (DOs) has been presented by (Kahn and Wilensky, 2006) and has been an important topic ever since. When talking about Digital Objects (DOs), generally any piece of information like a bit-sequence of data which has some metadata and is uniquely referenced by a PID is meant as defined by (Gary Berg-Cross et al., 2015). These DOs can be represented as collections which are relating to each other, e.g. some data which has some metadata and a PID attached to it. As stated in (Schultes and Wittenburg, 2019), the concept of FAIR DOs is a continuation of the DOs by putting them into the FAIR perspective. The goal is to extend the simplicity of the DO and make it possible to specify necessary domain-dependent metadata.

2.3 Linked Data

As the annotation of research data with their metadata is an important requirement presented by the use case, we briefly describe how metadata can be represented. A lot of work has been done in representing such information with the Resource Description Framework (RDF) (Wood et al., 2014) that structures information as subject, predicate, and object triple pairs which linked data builds upon. The main benefit is the annotation according to standards like ontologies (concept representations) which describe the meaning of a predicate like "dcterms:title". Therefore, this makes it a powerful tool to describe information concise but descriptive and machine-readable as linked data. Making these annotations accessible furthermore makes these types of metadata linked open data. However, in the use case the raw annotation of metadata alone is not enough since, especially for user-provided metadata, the need for validation emerges to make sure a certain structure is being followed. Such a mechanism and definition for validation is the W3C's Shapes Constraint Language (SHACL) (Kontokostas and Knublauch, 2017) standard, which enables the validation of metadata represented as linked data by comparing them to a validation schema.

2.4 Linked Data Platform - LDP

A Linked Data Platform (LDP) (Arwe et al., 2015) is a standard to model the interactions of web resources. It achieves this by proposing a simple interface based on HTTP operations that communicate linked data. Since a research data management system has to model the interaction between web resources and store related metadata linked to it, the LDP makes a perfect implementation target. Such an implementation is called an LDP server and differentiates between two types of so-called resources, the ones represented using RDF (Linked Data Platform RDF Source (LDP-RS)) and the ones dealing with the different types of data, not described in RDF, like simulations, image scans or test runs (Linked Data Platform Non-RDF Source (LDP-NR)). The definition of an LDP furthermore proposes the idea of a Linked Data Platform Container, which contains a number of resources and other containers. Since such a structure can be seen as related to an existing ontology like DCAT, there was work on aligning these two with each other, where the LDP can be utilized for the API definitions and DCAT to describe the hierarchical structure. The discussion can be followed in the respective issue in the GitHub repository: https://github.com/w3c/dxwg/issues/254.

2.5 FactStack

FactStack (Gleim et al., 2021) acts as an interoperable way for data management and provenance based on the FactDAG (Gleim et al., 2020) data interoperability layer model and is utilizing existing standards like the LDP and the HTTP Memento protocol. With this, they enable the version-based provision of resources with their metadata tied to it. This is done by assigning every resource with a FactID that can be resolved, thanks to time-based versions. They verify their concept with a reference implementation, showing real-life capabilities.

2.6 Solid

A recent work in progress in architectural standards is Solid (Capadisli et al., 2021) which acts as a specification for letting people store their data in so-called "Pods" that act as a data store. Solid enhances other standards like LDP and builds on top of their features like access rights and role management. Their vision is to have open and interoperable standards to correct the current notion of proprietary and diverging non-interoperable implementations.

2.7 Digital Object Architecture - DOA

The Digital Object Architecture (DOA) (DONA Foundation, 2019) is a specification by the DONA Foundation which defines information management standards and interfaces for interacting with Digital Objects (DOs). They define with their specification the DO itself, the Digital Object Interface Protocol (DOIP), the Digital Object Identifier/Resolution Protocol (DO-IRP), an Identifier/Resolution System, a Repository System and a Registry System. With this, they aim to create an interoperable infrastructure for data management.

2.7.1 Digital Object Interface Protocol - DOIP

The Digital Object Interface Protocol (DOIP) (Kahn et al., 2018) is a part of DOA and an interface definition on how to interact with a DO. It defines operations such as the creation, update, deletion, retrieval or search for it and specifies how the request and response should look like. In the later sections, DOIP will be seen as part of DOA and DOA will be evaluated as a whole.

2.8 FAIR Digital Object Framework - FDOF

The FAIR Digital Object Framework (FDOF) (da Silva Santos, 2021) is a currently developed framework for representing FAIR Digital Objects (FDOs) in a digital environment. The main goal is the representation of a DO according to the FAIR principles by enabling persistent identification, description with metadata records and providing their own ontology. For accessing the DOs, the FDOF extends the LDP structure and references the DOA. The main idea is that a defined identifier record has to exist which on the access of a persistent identifier is returned and always specifies all necessary information about the DO in a standardized way. Furthermore, a resolution protocol extending HTTP is proposed that can retrieve the identifier or metadata records with methods like "GETMETADATA".

2.9 Data Catalog Vocabulary - DCAT

The Data Catalog Vocabulary (DCAT) (Browning et al., 2020) is a recommendation of the W3C for describing the interoperability between data catalogs which act as collections of data. It standardizes their description and provides ways to describe datasets and their relationship to each other. Furthermore, it is possible to describe the related data services and the association to specific agents responsible for a dataset.

2.10 FAIR Data Point - FDP

The FAIR Data Point (FDP) (Bonino et al., 2021) is a service for FAIR metadata following the FAIR Guiding Principles and provides its own vocabulary. With their reference implementation, they define a central way for accessing metadata and define their interfaces based on standards like LDP. Their vocabulary definition extends DCAT by specifically specifying metadata, a metadata service and their own FAIR Data Point. They additionally utilize metadata schemas formulated by SHACL (Kontokostas and Knublauch, 2017) and include a standardized way of referencing them.

3 APPROACH

For this section, the use case requirements were collected in the context of implementing a standard-based research data architecture. It will be discussed how these requirements will be evaluated and what the different dimensions being looked at are.

3.1 Use Case Requirements

The platform Coscine offers researchers the ability to store their research data on several service providers, with the promise of being able to annotate them with metadata and persistently identify them using a PID service. This ensures the encapsulation of the research data management in the whole research data life-cycle, from planning to publication. Therefore, many things like the research data's location have to be accounted for to enable this at every step of the life-cycle and implemented standards and the provided interfaces need to fit into parts of the existing functionality. Therefore, the requirements are coming from the current platform's abilities, consider aspects of the FAIR Principles and the plans that the developing team is currently working on.

Requirement 1: The interface should be able to describe the data's and metadata's location independently of their physical storage location. Since research data has to be stored across multiple storage providers, this can pose quite a challenge, if not careful. Therefore, a standard that can deal with such a structure is generally favored.

Requirement 2: The standard should incorporate persistent identification. This means that research data should be made resolvable and annotated with such an identifier. The requirement is, additionally, one of the FAIR Principles for findability and accessibility.

Requirement 3: The standard should incorporate the annotation of research data with metadata as linked data. Here it is to note, that this annotation in the best case should be able to account for different levels of metadata, like descriptive, technical or administrative which might be located in different places. Fulfilling this requirement is furthermore increasing the interoperability of such a platform.

Requirement 4: The standard should provide a possibility to describe a research data management system's infrastructure. The need comes from the case that the use case platform not only describes and contains singular research items, but deals with whole research projects, which can contain multiple research data resources that can access separate storage providers. Such a structure should be possible to be described so that easy access to research data on every level can be established.

Requirement 5: The interface and standard should handle access rights. The structure of projects, resources and research data requires the possibility of managing access to every level separately since e.g. someone could become a project member and can see everything, but there might be the need to just share a certain part of the research data. This is envisioned to in turn enable collaboration with distributed read/write rights.

Requirement 6: The interface and standard should provide a way to manage data provenance information. For facilitating the reusability in a platform, data provenance is a key topic, so the option to describe the relations between separate versions of research data and describing the path research data has traversed is essential. Therefore, a standard should be able to account for separate versions of data and importantly also metadata, since they are subject to change as well.

Requirement 7: The standard should provide a clear and standardized API. For making the platform in line with accessibility requirements, it is necessary that a standardized protocol and interfaces can be used to communicate research data and their metadata.

Requirement 8: The standard should be in a state usable for production. This means the standards should be well-supported by a community, established and in a production-ready version. This is to ensure the maturity of the standard and platform.

3.2 Methodology

Since the requirements of the use case are clear, it needs to be discussed how the standards shall be evaluated regarding them. From the descriptions in section 3.1, the following categorization of them can be made:

1. Description of the Data's Location
2. Handling of Persistent Identifiers
3. Metadata as Linked Data
4. Description of the Structure
5. Handling of Access Rights
6. Description of Data Provenance
7. Clear Standardized APIs
8. Ready for Production

These numbered categories will be used and checked for every standard which include LDP, FactStack, Solid, DOA, FDOF, DCAT and FDP as presented in section 2. They are ranked into either conforming (+), semi-conforming (/) or not conforming (-). Since it is expected, that no single standard will fully account for all the requirements, it will be looked into, how they can be combined and what their compatibility to each other are. Thankfully, most of them have a similar base (derived from LDP), which should make this possible and is visualized in figure 1.

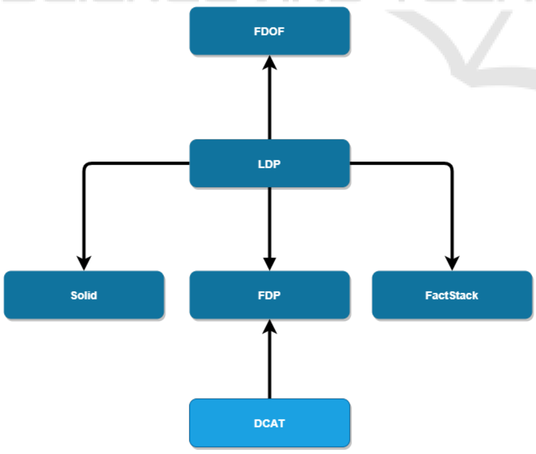

Figure 1: Illustration of the standard relations.

4 EVALUATION

In this section, the previously described architecture standards are compared with each other according to the requirements brought forward by the use case in section 3. Furthermore, the results will be discussed and the plans moving forward from that are described. Each of the standards is designed to meet a specific purpose, some of them with FAIR principles in mind (like FDP) some of them not (LDP). All of them are fit to serve their designed purposes, the question that is to be answered is if they can serve the use case at hand (becoming the standardized interface for Coscine) according to the previously discussed requirements.

4.1 Comparison

The full evaluation of the requirements can be found in table 1. This subsection will elaborate on each requirement and the ranking of the individual standards.

Table 1: Evaluation of the requirements listed in section 3.2.

Features	1.	2.	3.	4.	5.	6.	7.	8.
LDP	-	-	+	/	-	-	+	+
FactStack	-	+	+	/	-	+	+	+
Solid	-	-	+	/	+	/	+	+
DOA	-	+	-	/	/	-	+	+
FDOF	-	+	+	+	/	/	/	-
DCAT	+	-	/	/	/	/	-	+
FDP	+	-	+	+	+	/	+	+

4.1.1 Description of the Data's Location

Starting with the comparison between the standards, the first requirement to look at is the possibility to describe the research data's location. While evaluating the standards, LDP, FactStack, Solid, FDOF and DOA left the handling of the data's location open and focus more on the interfaces which make the data available that an implementing agent has to provide. DCAT and with that respectively FDP describes a clear model on how data exposure could be described and with that offers a way to describe so-called data services that can contain information like an endpoint URL which points directly to where the data is located. The differences between the standards are displayed in table 1 as "1.".

4.1.2 Handling of Persistent Identifiers

For handling PIDs, LDP, Solid, DCAT, and FDP all expect identification of data, but do not fully provide a direct solution on how to create and manage PIDs in their standards. That being said, they do not prevent one from using persistent identifiers as an identifier solution, either. FactStack, DOA, and FDOF on the other hand specifically utilize persistent identifiers in their definitions and therefore fulfill this category.

The differences between the standards are displayed in table 1 as “2.”.

4.1.3 Metadata as Linked Data

The handling and description of metadata as linked data, tied to their research data, marks a significant aspect that most standards support. LDP, FactStack and Solid enable this with their LDP-RS type and enable the description of LDP-NR types with the “describedby” property. FDP extends DCAT by exactly this metadata dimension, however DCAT itself can describe some parts of the metadata already, since the technical metadata description is one of its primary goals. Additionally, FDOF incorporates the metadata part by establishing a concrete metadata record that can be accessed and has to be described in RDF. The only standard which does not support the requirement directly is DOA, since it relies heavily on its own protocol for communication. It does, however, take metadata into account, leaves the implementation part, however, quite vague. The differences between the standards are displayed in table 1 as “3.”.

4.1.4 Description of the Structure

For describing the complex use case’s structure, most standards offer some solutions. The LDP-based solutions with FactStack and Solid all can offer a structure description by abstracting it to the container model. DCAT comes close to being able to describe the whole structure and is an excellent fit but falls short on the metadata part where FDP comes in and completes the picture by furthermore including parts of the LDP. Following the aims of FDOF, it would also help to incorporate the full structure by presenting a solution for describing the identifier, metadata and data interaction and relying in parts on the LDP. DOA provides with its three core components, the identifier/resolution system, the repository system and the registry system, fitting components for the structure. Without some clear definitions about them in the terms of the linked data space, it is however not fully adaptable for the use case. The differences between the standards are displayed in table 1 as “4.”.

4.1.5 Handling of Access Rights

With a focus on open data and metadata, LDP and FactStack do not provide clear access rights handling. For this, Solid comes in and provides a way to handle access rights on a granular level, which can enable collaboration and sharing of data. DOA’s DOIP standard provides a method for sharing access control information, but the usage is not entirely specified. FDOF specifically mentions a metadata access mechanism and leaves the implementation of access mechanisms open to the implementing platform. DCAT offers definitions on how to define access rights, which FDP expands on by providing its own access control components, handling the access to metadata sets. The differences between the standards are displayed in table 1 as “5.”.

4.1.6 Description of Data Provenance

The LDP standard is not concerned with versions and provenance, which is why FactStack fills this role. Solid mentions provenance as a notion to be stored as auxiliary resources, nevertheless no concrete concept can be found. FDOF mentions provenance in their working draft, nevertheless the work is not quite complete yet. With DCAT and FDP, there is a clear definition on how to describe provenance information for datasets, the interaction with it is, however, missing. The DOA does not have a clear definition on provenance, theoretically could be, however, extended to be utilized by accessing a defined version of a digital object. The differences between the standards are displayed in table 1 as “6.”.

4.1.7 Clear Standardized APIs

For providing clear APIs, most standards provide their own definitions. The only standards which do not currently are FDOF due to the draft status and DCAT because it does not aim to provide an API. The differences between the standards are displayed in table 1 as “7.”.

4.1.8 Ready for Production

The readiness for production was fulfilled by nearly all standards except FDOF, which because of its draft status and no reference implementations is right now not deemed as ready for production. It is, however, still very relevant because active work on it could change this. The differences between the standards are displayed in table 1 as “8.”.

4.2 Discussion of the Results

From looking at the comparisons evaluated in 4.1, two kinds of needs were identified: Defining the structure and defining the access and API endpoints. It seems that for each category, there is a top contender for a baseline. For defining the structure, this would be DCAT because of a wide array of definitions for defining data catalogs, datasets and their distribution information. For defining access and API endpoints, the baseline is LDP, since most other definitions, except DOA, are based on this standard. This makes it clear that these two standards are the most common ground to which the use case should be compatible, and there is even some alignment between them. However, the baselines do not offer the full requested requirements, making the extensions worthwhile. Especially, the extensions of LDP as FactStack and Solid can bring a lot of wanted functionality in terms of data provenance and access rights which because of the common ground of being based on LDP is not that big of an additional overhead to implement and is definitely something the use case will strive to be compatible to. Additionally, since FDP extends the LDP to become a metadata access point and extends DCAT with the part of recognizing metadata as its own entity, this compliments the final requirements and is therefore a final piece for achieving the requirements. However, DOA is not completely out of the picture either, since especially for DOA, the persistent identifier resolution part which DONA (the maintainer of DOA) is working on is based on the Handle System which the use case uses as well. There are, therefore, certainly parallels, so this is something not to disregard and in future some compatibility is to be expected. Lastly, FDOF is an interesting current development which because of the not production readiness just falls short currently to be implemented. This, however, could definitely change in a short amount of time, making this a future candidate to look out for.

5 CONCLUSION

This paper discussed the need for standards in a research data management system and presented the use case of Coscine, which acts as such a platform. Common architectures and standards are explored and described, including LDP, FactStack, Solid, DOA, FDOF, DCAT, and FDP. For evaluating these standards, the requirements of the use case are discussed and presented. During the evaluation, the standards are categorized regarding presented requirements. The discussion part clarifies that there is a baseline which many standards fall back on, which is LDP and DCAT. They are therefore a definite must for the use case to be compatible to. To fulfill the requirements, FactStack, Solid, and FDP are discussed to fulfill the missing parts from the baseline. Therefore, after evaluation, work on implementing these standards in the use case can start, and it can hopefully become a fully standard-based research data management system in the future.

REFERENCES

Arwe, J., Malhotra, A., and Speicher, S. (2015). Linked data platform 1.0. W3C recommendation, W3C. https://www.w3.org/TR/2015/REC-ldp-20150226/.

Bonino, L. O., Burger, K., and Kaliyaperumal, R. (2021). Fair data point - working draft, 23 august 2021. https://specs.fairdatapoint.org/.

Browning, D., Cox, S., Beltran, A. G., Albertoni, R., Winstanley, P., and Perego, A. (2020). Data catalog vocabulary (DCAT) - version 2. W3C recommendation, W3C. https://www.w3.org/TR/2020/REC-vocab-dcat-2-20200204/.

Capadisli, S., Berners-Lee, T., Verborgh, R., and Kjernsmo, K. (2021). Solid Protocol. Version 0.9.0, 2021-12-17, W3C Solid Community Group. https://solidproject.org/TR/protocol.

da Silva Santos, L. O. B. (2021). Fair Digital Object Framework Documentation - working draft. https://fairdigitalobjectframework.org/.

DONA Foundation (2019). Digital Object Architecture. https://www.dona.net/digitalobjectarchitecture.

Gary Berg-Cross, Raphael Ritz, and Peter Wittenburg (2015). Rda dft core terms and model.

Gleim, L., Pennekamp, J., Liebenberg, M., Buchsbaum, M., Niemietz, P., Knape, S., Epple, A., Storms, S., Trauth, D., Bergs, T., Brecher, C., Decker, S., Lakemeyer, G., and Wehrle, K. (2020). Factdag: Formalizing data interoperability in an internet of production. *IEEE Internet of Things Journal*, 7(4):3243–3253.

Gleim, L. C., Pennekamp, J., Tirpitz, L., Welten, S. M., Brillowski, F. S., and Decker, S. J. (2021). FactStack : Interoperable Data Management and Preservation for the Web and Industry 4.0. In *Datenbanksysteme für Business, Technologie und Web (BTW 2021) : 13.-17. September 2021 in Dresden, Deutschland / K.-U. Sattler et al. (Hrsg.)*, volume 311 of *GI-Edition. Proceedings*, pages 371–395, Bonn. 19. Fachtagung für Datenbanksysteme für Business, Technologie und Web, online, 19 Apr 2021 - 21 Jun 2021, Köllen. Konferenzort: Dresden, Germany. - Datenträger: CD-ROM. - Weitere Reihe: Lecture Notes in Informatics ; 371.

Jacobsen, A., de Miranda Azevedo, R., Juty, N., Batista, D., Coles, S., Cornet, R., Courtot, M., Crosas, M., Dumontier, M., Evelo, C. T., Goble, C., Guizzardi, G., Hansen, K. K., Hasnain, A., Hettne, K., Heringa, J., Hooft, R. W., Imming, M., Jeffery, K. G., Kaliyaperumal, R., Kersloot, M. G., Kirkpatrick, C. R., Kuhn, T., Labastida, I., Magagna, B., McQuilton, P., Meyers, N., Montesanti, A., van Reisen, M., Rocca-Serra, P., Pergl, R., Sansone, S.-A., da Silva Santos, L. O. B., Schneider, J., Strawn, G., Thompson, M., Waagmeester, A., Weigel, T., Wilkinson, M. D., Willighagen, E. L., Wittenburg, P., Roos, M., Mons, B., and Schultes, E. (2020). FAIR Principles: Interpretations and Implementation Considerations. *Data Intelligence*, 2(1-2):10–29.

Kahn, R. and Wilensky, R. (2006). A framework for distributed digital object services. *International Journal on Digital Libraries*, 6(2):115–123.

Kahn, R. E., Blanchi, C., Lannom, L., Lyons, P. A., Manepalli, G., Tupelo-Schneck, R., and Sun, S. (2018). Digital Object Interface Protocol Specification. https://www.dona.net/sites/default/files/2018-11/DOIPv2Spec_1.pdf.

Kontokostas, D. and Knublauch, H. (2017). Shapes Constraint Language (SHACL). W3C recommendation, W3C. https://www.w3.org/TR/2017/REC-shacl-20170720/.

Politze, M., Claus, F., Brenger, B. D., Yazdi, M. A., Heinrichs, B., and Schwarz, A. (2020). How to Manage IT Resources in Research Projects? Towards a Collaborative Scientific Integration Environment. *European journal of higher education IT*, 1(2020/1):5.

Schmitt, R. H., Anthofer, V., Auer, S., Başkaya, S., Bischof, C., Bronger, T., Claus, F., Cordes, F., Demandt, É., Eifert, T., Flemisch, B., Fuchs, M., Fuhrmans, M., Gerike, R., Gerstner, E.-M., Hanke, V., Heine, I., Huebser, L., Iglezakis, D., Jagusch, G., Klinger, A., Krafczyk, M., Kraft, A., Kuckertz, P., Küsters, U., Lachmayer, R., Langenbach, C., Mozgova, I., Müller, M. S., Nestler, B., Pelz, P., Politze, M., Preuß, N., Przybylski-Freund, M.-D., Rißler-Pipka, N., Robinius, M., Schachtner, J., Schlenz, H., Schwarz, A., Schwibs, J., Selzer, M., Sens, I., Stäcker, T., Stemmer, C., Stille, W., Stolten, D., Stotzka, R., Streit, A., Strötgen, R., and Wang, W. M. (2020). NFDI4Ing - the National Research Data Infrastructure for Engineering Sciences.

Schultes, E. and Wittenburg, P. (2019). Fair principles and digital objects: Accelerating convergence on a data infrastructure. In Manolopoulos, Y. and Stupnikov, S., editors, *Data Analytics and Management in Data Intensive Domains*, pages 3–16, Cham. Springer International Publishing.

Schwardmann, U. (2015). epic persistent identifiers for eresearch. In *Presentation at the joint DataCite-ePIC workshop Persistent Identifiers: Enabling Services for Data Intensive Research, Paris*, volume 21.

Sun, S., Lannom, L., and Boesch, B. (2003). Handle system overview.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., and Evelo, Chris T. ... Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, 3:160018.

Wood, D., Lanthaler, M., and Cyganiak, R. (2014). RDF 1.1 Concepts and Abstract Syntax. W3C recommendation, W3C. https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.