EPiC Series in Computing

Volume 109, 2026, Pages 133–144

Proceedings of EUNIS 2026 Annual Congress

Designing the Architecture of an Ecosystem for the Management of Research Data and Software

Jan Bernoth[1*], Andreas Hartmann[2+] and Ulrike Lucke[3*]
[*] University of Potsdam, Germany
[+] Leipzig University of Applied Sciences, Germany
jan.bernoth@uni-potsdam.de, andreas.hartmann@htwk-leipzig.de,
ulrike.lucke@uni-potsdam.de

Abstract

Establishing open science practices faces the challenge of integrating new workflows and tools with scientific infrastructure grown for decades. Thus, initiatives providing new services for research data and software management have a strong need to establish a flexible and sustainable system architecture, covering both existing and emerging parts, and at the same time being open for further refinement. Based on established principles and solutions as well as a requirements analysis with the community, this article provides a distributed architecture to cope with these challenges. Added value is demonstrated based on facilitated use cases, and further directions of work are lined out.

1 Evolving Infrastructure for Open Science

The FAIR principles (findable, accessible, interoperable, and reusable) for research data (Wilkinson et al., 2016) and software (Chue Hong et al., 2022) demand action from three groups: *researchers*, who must adhere to the principles to produce FAIR research data and software; the *research discipline community*, which must create specific vocabularies and standards for interoperability; and *research infrastructure providers*, who must construct a user-friendly infrastructure to exchange, reproduce, and archive research outputs. Research software has also been recognised as an indispensable component behind data (Barker et al., 2022), resulting in data plus software requiring an execution environment to be preserved as executable research results (Goedicke & Lucke, 2022). This opens up the field of research data and software management (RDSM). Compared to "just" data, the involved systems and processes are more complicated, management is more complex, and necessary competences are widened (di Cosmo, 2020; Lucke, 2022).

[1] https://orcid.org/0000-0002-4127-0053
[2] https://orcid.org/0000-0003-1340-5325
[3] https://orcid.org/0000-0003-4049-8088

L. Desnos, C. Diaz, J. Mincer-Daszkiewicz, L. Merakos, R. Vogl, S. McLellan and U. Lucke (eds.), EUNIS 2026 (EPiC Series in Computing, vol. 109), pp. 133–144

From a technical perspective, a FAIR ecosystem is created in which different pillars are supporting possible action in RDSM, e.g. sharing and curating research output and teaching skills (L'Hours et el., 2022). The implementation of these components has to be done on institutional, disciplinary, national and international level, which is currently supported by several initiatives. The different approaches from these initiatives to construct a technical infrastructure for their ecosystems, like the *Gaia-X Architecture* (Gaia-X, 2025) or the *EOSC Architecture and Interoperability Framework* (Williams et al., 2023), indicate that this is a highly heterogeneous, networked landscape of services that are rather subject to feudal than federal, coordinated governance archetype (Weill, 2010). Navigating this ecosystem demands a high focus on the interoperability of the solutions used. Interoperability has been discussed at various levels to date, e.g. the definition of standardised exchange formats (for data and metadata), centralised docking points (e.g. repositories or identity providers) or predefined processes (e.g. for quality assurance). Since these discussions are still ongoing and no unified framework has yet emerged, any new technical implementation must be designed with both flexibility and scalability in mind, capable of adapting to evolving standards, while at the same time grounding itself in those broader concepts and principles that have already reached sufficient consensus to be considered stable.

As a consortium for a specific discipline and within national scope, NFDIxCS (National Research Data Infrastructure for and with Computer Science) addresses this challenge from a computer science perspective in Germany. The blueprint focuses on preserving research artifacts in a Research Data Management Container (RDMC) (Goedicke & Lucke, 2022), combining tools and platforms for the joint management of data and software. Beyond preservation, the artifacts must remain usable within a FAIR research data and software lifecycle, supported by dedicated services that enable finding and reusing artifacts, as well as extending established processes of publication. This article presents the resulting architecture as a contribution to that effort. Though originating from one discipline, applicability to other fields and international connectivity are considered from the very beginning.

The remainder of this article is structured as follows. Chapter 2 provides an overview of previous and related work. Chapter 3 describes the immediate prerequisites and methods of our work. Chapter 4 presents the resulting architecture. Finally, a summary and outlook are given in chapter 5.

2 Related Work

Since the developed solution requires integration into an existing research IT ecosystem, a generic analysis of its context of use is necessary. For this purpose, the requirements engineering framework by Pohl (2010) is applied. It divides the ecosystem's context into four areas, referred to as facets.

- The *subject facet* concerns relevant objects and events in the context of a system (Pohl, 2010). For RDSM, the subject is research data and software, in our case compounded in an RDMC (Goedicke & Lucke, 2022). Events are triggered by the RDSM lifecycle, i.e. publishing & sharing, preserving, and reusing steps must be addressed. For publishing & sharing, established scholarly practices for curating and evaluating research output have to be supported, and compatibility with standardized formats is required. Metadata standards such as DataCite, schema.org, and Dublin Core have long been established for data, while the standardisation of data formats remains limited to selected disciplines. For software, relevant standards are still evolving, including CodeMeta (Jones et al., 2023), Citation.cff (Druskat et al., 2024), and classification schemata for research software (Hasselbring et al., 2025). For preservation, compatibility with these standards remains crucial for long-term accessibility. In case of reusing, additional support, for instance by semantic search technologies (Ateia, 2025), enable others to find, understand, and reuse the artifacts.
- The *usage facet* focuses on the interaction with users or other services in order to achieve a goal or accomplish a task (Pohl, 2010). For users, there is a significant overlap with the subject

facet, as users need to publish, share, curate, and preserve their research data and software. Since FAIR does not necessarily imply open access, a system must support differentiated access permissions for artifacts. Regarding interaction with other services, interoperability with the broader FAIR ecosystem is essential. Users and institutions have created their own research IT environments, resulting in the publication process to be handled at least partly by third-party services. Published RDMCs must be indexable and searchable by other services as well. Moreover, existing Codes of Conduct (like DFG, 2025) and legislation, such as the General Data Protection Regulation (GDPR), have to be obeyed.

- The *IT system facet* requires consideration of the technical and operational environment, including relevant policies and strategies (Pohl, 2010). This facet is the primary perspective addressed in this article. The technical environment is shaped by several platforms available either for specific disciplines, such as Pangaea for earth and environmental science or EMBL-EBI for biology, and general-purpose platforms such as Zenodo or GitHub. These repositories represent de-facto standards in their respective fields and often act as a driving force for further development. Consequently, interoperability with these existing platforms and services is key. Since globally interconnected science is incompatible with centralised operating and governance structures (Möller et al., 2025), the ecosystem must operate within distributed ecosystems with mutual trust (Marsh & Dibben, 2003). However, the development of an IT architecture in the true sense (Ahlemann et al., 2012) is rarely to be seen yet. There are examples available from related fields, like the Higher Education Reference Model (CAUDIT, 2025) where RDSM is covered as a single business capability. More detailed architectural models are available for education processes (Benzinger et al., 2025; Hartmann et al., 2025). In the field of research, especially RDSM, the available architecture specifications are so far limited to EOSC. On a national level, the German NFDI initiative is working on this topic, but has a focus on research data and does not yet cover research software as an integral part. This gap is addressed here.
- The *development facet* addresses all aspects of the context related to the development process, such as process guidelines, quality assurance methods, and maturity models (Pohl, 2010). Several research groups are on their way to create sustainable architecture descriptions, often via descriptions of personas and use cases (Schmitt et al., 2020; Manzke & Lorenz, 2024; Bernoth et al., 2024). The resulting use cases demonstrate the strong requirements for interoperability between different services across the research community. Development is part of an ongoing scientific context, meaning that there is no dedicated development team, but rather researchers creating RDSM prototypes for their research. Therefore, the architecture must structure the communication between these prototypes.

The generic analysis of the context of use for the ecosystem shows how the connections to other systems and the user scenarios shape the system, without defining research data and software in detail. Taking this into account when weighting the quality criteria according to ISO/IEC 25010, the system should focus on *Functional Suitability* and *Reliability* to address the usage and subject facets, on *Security* to address the IT system facet, and on *Flexibility*, which is relevant across all facets.

3 Methods Used for Development of the Architecture

Our methodological approach is described in section 3.1, followed by an explanation of the general architectural model we applied in section 3.2, and an overview of the structure and services of the RDSM ecosystem to be considered in section 3.3.

3.1 Iterative Approach

To develop a reference architecture of an RDSM ecosystem, we combined a user centric perspective, including use-cases based on demands by the scientific community with a management view in order to harmonize vocabulary, structures, processes or conflicting positions. Based on existing personas and use cases, which have been collaboratively designed in a series of community workshops (Bernoth et al., 2024), representatives of the sub-communities then worked out more detailed specifications of how to implement the respective RDSM use cases. We provided them with a) the existing use case descriptions, b) the overall objectives of the initiative, and c) a layered architecture reference model strengthening the technical implementation perspective including security aspects. This model is explained in the next section. Participants assigned existing and new components to realize the use cases and then placed them in the architecture blueprint, thus creating drafts of architectural segments specific to these use cases.

After the workshop, the resulting descriptions of how to implement these 15 use cases were further processed in an iterative approach. After translation to a digital format, consent on proper translation was requested from the authors. This was followed by the super-position of all drafts. For this, we identified recurring components, and we classified them regarding their position in one of the layers and in a specific administrative domain. In some cases, we had to further break down some components that had been sketched only roughly. In other cases, changes had to be made regarding naming conventions or involved communication paths.

As a result, a combined architecture description ready to implement all covered use cases was available. This was in turn discussed with the original authors of the use case descriptions to check for consistency with their intentions and demands. Afterwards, the resulting architecture blueprint was discussed with the representatives of the sub-disciplines of the German Informatics (GI) association to gain approval on a higher abstraction level. Feedback only concerned comprehension questions and minor suggestions for implementation, thus resulting in overall consent.

Throughout this process, we continuously consulted a number of experts in the field of RDSM, IT architecture and IT security to ensure the validity of our results.

3.2 Applied Layer Model (Architecture Blueprint)

Our core architectural design is based on a network zoning model defined by Schönbächler & Pfister (2017), which was transformed into a layered component architecture by Hartmann et al. (2025) and successfully applied to specify the architecture blueprint of a national digital education ecosystem (NDEE). The zoning model was limited to a network perspective and suffered from modern security concepts, such as Zero Trust. The layered NDEE approach is advanced in many aspects. It describes a whole user-centred digital ecosystem including many services and components from various domains. Network rules between the zones do still apply (e.g. HTTPS). However, now each application must implement a component-oriented architecture pattern, while its components will be assigned to the appropriate layers (wrapped within services). For example, a GUI component cannot be located in inner/lower layers. Instead it will be provided as a service through a platform (e.g. portal) above and does securely communicate with backend (data) services. As a result, the NDEE ecosystem uses a well defined service integration architecture, based on the five layers. In addition, Zero Trust Core Principles (Ghosh et al., 2021) apply between, as well as within these layers. As per definition, any service is an asset with an unambiguous identity in the ecosystem, and is never trusted – but always verified. The transformed network zones (or component layers) are defined as follows:

- *Entry Zone*: a public-facing gateway where traffic first enters the network. Usually, a web application firewall (WAF) secures connections coming from the internet. From a user's perspective, a landing page may be the first point of contact, including access control and traffic management. The entry zone shall be the only path to enter the network.

- *Application Zone*: contains all web applications with focus on application logic and user interfaces. Typically, a web application portal provides the technical platform for that. Providing basic features, these applications shall call backend services for (sensitive) business logic or data. Note that backend services shall implement their administrative or management access as GUI components in this zone, too, and connect them via service requests. The application zone should not host any data.
- *Service Zone*: a secure area where backend services and microservices reside, handling business logic that the web applications above call upon. Interface standards shall include RESTful and SOAP APIs and make use of validatable XML or JSON. Non-sensitive data storage may be assigned to this zone.
- *Integration Zone*: a dedicated zone for connecting with various enterprise ecosystems through appropriate middleware, such as an ESB. The zone shall manage any integration with third-party APIs or other enterprise applications, especially containing and providing sensitive data. The zone shall realize data synchronization and transformation (ETL).
- *Secure Hosting Zone*: a highly secure environment that shall only be accessible though middleware and connectors from the integration zone. It typically contains backend services storing sensitive data or sensitive third-party enterprise applications, such as components being subject to GDPR surveillance.

This five-layered architecture is a basis to implement a secure IT ecosystem adding zero-trust by allowing connections only to neighboured components either within the same layer (maybe in different administrative domains) or to layers immediately above or below (within the same domain).

3.3 Contextual Factors

In computer science, RDSM encompasses very diverse research software and data lifecycles, meaning that research artifacts can be composed of multiple components that are functionally dependent on each other. Additionally, as software is always involved in handling the data, there is a need to archive the executional runtime environment itself to preserve not only the files but also the reproducibility of the experiment. One solution could be a container-based approach that encapsulates data, software with its contextual information, and its execution environment (Goedicke & Lucke, 2022). To support the creation of such a research data management container (RDMC), a wizard was created (Bernoth et al., 2025) and will be further developed to streamline and automate the process of capturing, structuring, evaluating, and publishing research software and data in compliance with FAIR data principles. The creation of the RDMC includes the construction of a Remote Execution Environment (REE) in which the runtime environment is built and archived to effortlessly reproduce the research results (Blessing et al., 2025). Although both concepts are still in the prototype phase, RDMC, with the REE as one of its essential components, is a core part of the resulting architecture. It is surrounded by a set of dedicated services, provided via the NFDIxCS API Manager.

In addition, there exist several dedicated platforms for data and software archiving such as Zenodo, DataVerse, GitHub or Software Heritage Archive, providing enhanced features for software archival and referencing (di Cosmo, 2020). For computer science, the dblp bibliography is important for publications and referencing. These platforms also appear as services in the architecture, partly in conjunction or in competition with institutional storage services and emerging RDSM services such as GitLab, Jupyter, electronic lab notebooks etc. (Rans & Whyte, 2017). Moreover, researchers use a plethora of tools and data storage in their personal IT environment, not to mention self-developed software, which also has to be connected to the overall infrastructure to ensure smooth transition processes (Treloar & Klump, 2019).

Basic functionality such as authentication & authorisation, persistent identifiers or semantic technologies, like knowledge graphs or terminology, are not provided specifically for computer science, but are made available throughout Germany by the NFDI as basic services for all disciplines (Politze et

al., 2025). With regard to the overarching context, it should be mentioned that NFDI is to act as an EOSC node, i.e. the services mentioned will be fundamentally compatible with the IT, organisational and governance structures emerging at European level (Williams et al., 2023).

To sum this up, Tab. 1 assembles the list of services to be considered, grouped by the respective administrative domains they belong to and a characterization of our potential influence on these services, their functionality and interfaces.

Domain	NFDIxCS core	NFDIxCS services	NFDI basic services	Institutional services	Personal research environment	Community services
Influence	full	full	partly	limited	limited	rare
Services	API Manager	Portal, RDMC, REE, Anony-mization, Curation, Search, Artifact Evaluation, …	Identity & Access Management, Persistent Identifiers, …	GitLab, Jupyter, Electronic lab notebooks, Institutional cloud storage, …	Local storage, Local tools, Own research software, …	Zenodo, Dataverse, GitHub, Software Heritage Archive, Dblp …

Table 1: Administrative domains and selected services to be considered in the architectural model

Note that while technically there is no access to institutional or personal infrastructure, those can be addressed to some extent via community development approaches (negotiation processes; education and training). The list of services resp. components resulting from our community workshops, as indicated here, were then assembled towards an architecture description as described in the next chapter.

4 Results

The challenge in applying the architectural blueprint is to assign all NFDIxCS components required to implement the defined or aspired use cases to a certain a) layer as defined in subsection 3.2, and b) administrative domain as given in Tab. 1. After placement of all components, verification is needed if the workflow associated with the use cases can be processed without violating the security rules of connecting only to immediate neighbors within any layer. This may result in a relocation or re-design of services or components, or even in a partial redesign of workflows or use cases. The result of this iterative adjustment is the architecture presented below.

4.1 Architecture

The architectural approach is a combination between centralized and decentralized elements. The entry point for different user groups is the *website* (after passing the WAF), which guides users after login into the NFDIxCS Portal. The portal has an identity & role-based concept that allows privileged users to interact with research artifacts in different roles. In the service zone, a microservice architecture approach supports the flexible, decentralized development of services along the research artifact lifecycle. This aligns with the structure of the NFDIxCS consortium, which is also thematically split into different task areas concerning these aspects. The middleware components in the integration zone

provide the secure connection towards internal enterprise and data services, including self-hosted databases. It also connects to external and/or third-party services using API gateways and reverse proxies if necessary.

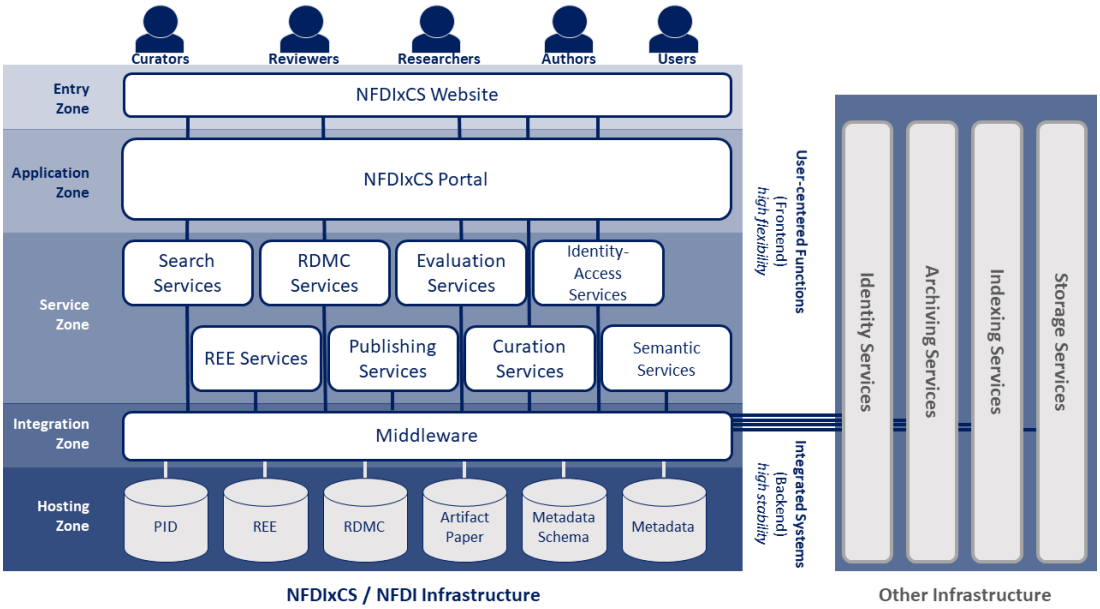

Figure 1: Proposed architecture of a RDSM ecosystem.

The core components of the architecture blueprint implementation provide the following functionality within the overall NFDI ecosystem:

- The *Website* is the entry point for any user group, such as curators, reviewers, researchers, authors, and data or software users. Beyond general information about the consortium and their activities, it provides access to the *Portal*. After authentication, users gain access to the Portal, which serves as a role-dependent workspace. Services and features are accessible depending on the task the user wants and is authorized to perform.
- General NFDI and dedicated NFDIxCS services represent the core business logic for interacting with the RDSM lifecycle (searching, creating, curating, and publishing). For clarity, they are grouped into the following clusters: *Search services* allow users to discover and find RDMCs. *RDMC services* cover creating, viewing, and deleting RDMCs. They are closely linked to *REE services*, as an REE is triggered by the RDMC services and can be executed by opening it. By design, REE services are not accessible directly from the Portal, since REEs should only be created with metadata provided by the RDMC. *Publishing services* build on RDMC services; once an RDMC is created, it can be published together with an artifact paper that describes its content in a human-readable way. Publications can also be directed to existing archiving (e.g., Zenodo, Dataverse, SoftwareHeritage), indexing (e.g., registries, knowledge graphs), or storage services (e.g., OpenCloud, Nextcloud). *Evaluation services* support scientific quality assurance, for instance through artifact evaluation for conferences or journals. *Curation services* enable data stewards or other support staff to connect, annotate, and manage datasets and research software. The *Identity-access services* manage identities and roles for granting access to services. Finally, the *Semantic services* provide mechanisms to add metadata to the artifacts and link them to existing semantic technologies.

- The *Middleware* in our solution acts as a connector between the NFDI services above and core data services in the secure hosting zone. Technically, it is a central API Manager through which every service and database, whether internal or third-party, can be connected and exchange information. By centralizing communication through a single layer, the Middleware simplifies integration and workflows, ensures consistent data exchange, enables traffic monitoring and logging, and allows new services or databases to be added without affecting existing ones.
- The *Data services* in the secure hosting zone store and provide the data created by the NFDI services. Importantly, this infrastructure can be used and hosted not only by NFDIxCS but also by other operators. Each data service contains a database and serves a specific purpose. The *PID* database stores persistent identifiers that link to *RDMCs* and *REEs*, which are in turn kept in their own dedicated databases. *Artifact papers* contain descriptions of the purpose, content, and usage of artifacts. *Metadata* and *Schema* holds not only descriptive metadata for RDMCs but also taxonomies, semantic information, and ontologies. *User/Roles* manages roles, security settings, and user profiles within the platform/ecosystem. The *Search indexes* support the search services in delivering fast and efficient search results.
- NFDIxCS services need to be connected to other services in the ecosystem. They can be clustered into four main topics. *Identity services*, such as ORCID, provide user information during the registration process. *Archiving services*, such as Zenodo, DataVerse, or SoftwareHeritage, are used to properly preserve data, software, and RDMCs over a longer period of time. *Indexing services* encompass semantic technologies, such as ORKG or Wikidata, which help users discover artifacts by connecting their embedded knowledge to broader knowledge representations. *Storage services*, such as Nextcloud, Dropbox, or OpenCloud, are similar to archiving services, with the key difference being that their purpose is focused on active data use rather than long-term preservation.

In the next section, we present sample use cases to illustrate how this architecture ensures the interoperability of new and existing services while addressing autonomy and security concerns.

4.2 Use Cases

This condensed presentation of the architecture necessarily omits details, particularly regarding how its components interact. To address this, we will present four core use cases that illustrate the interplay of the above-mentioned components.

Login and role-based access to the platform: The login process is similar across all following use cases and will therefore be described once here. Logging into the system requires credential validation, whereby an Identity Access Service authenticates the user via either the basic NFDI service IAM4NFDI or an external identity provider such as ORCID. Once the user has been validated, a role management service determines which roles the user can obtain and stores these, via middleware, in the role database. Based on their assigned role, the user can then see and access the available services in the portal. Where applicable, the user may also switch between roles, since some services have different entry points depending on the role selected.

Publishing data and creating an RDMC: After logging in via the NFDIxCS website, researchers can use the RDMC creation service within the NFDIxCS portal. While creating an RDMC, the researcher must provide metadata by selecting from a taxonomy presented by the semantic service or answering predefined questions. Once the artifacts have been compiled and enriched with metadata, the REE will be automatically created by an REE service. With all components ready, the RDMC and REE will first be signed by a dedicated RDMC service, and subsequently published by a publication service. Depending on the use case, the RDMC along with its REE will either be submitted for artifact evaluation or will be published as is. Finally, the publishing service will store the necessary files in its own hosting zone and forward the artifacts to external archiving services.

Reviewing submitted artifacts: To review an RDMC with its REE, the user must log in and select the reviewer role within the NFDIxCS portal. The reviewing service orchestrates several distinct activities, starting with verifying the validity of the RDMC signature via a dedicated RDMC service. Additionally, it will soon be possible to use an REE service to execute the REE in order to verify the reproducibility of the artifact. The evaluation service then initiates the review phase by creating a Cloud IDE for anonymous code review. Comments are subsequently collected for artifact review or rebuttal. After successful completion of the review phase, the RDMC service will automatically add the new metadata and badges to the RDMC, along with any updates to the code or dataset. This triggers a new automated RDMC and REE creation process, using the initially submitted RDMC as a basis and applying the updates accordingly. Finally, the signing service will be invoked, and the publishing service will guide the publishing process.

Searching for RDMCs: For this use case the user does not need to be logged in. If the user chooses to search without login, the NFDIxCS portal will automatically create an anonymous account in the background, allowing the platform to log search activity and continuously improve the search service. The search service will use a local database with indexed RDMC metadata to provide the user with optimized and fast search results. To ensure this, the search service updates the indexing database both periodically and automatically upon the publication of a new RDMC. After finding an RDMC, a dedicated RDMC service automatically verifies its validity, and the user may optionally choose to invoke an REE service to reproduce the results. For authenticated users, the search history can be stored for future reference.

Other use cases (for instance, on enriching metadata and configuring an RDMC service for artifact evaluation) exist, but are omitted here for the sake of brevity.

5 Discussion and Following Steps

In this article, we have added a differentiated architecture proposal for the management of research data and software to the existing reference models and architectures for higher education IT, which have so far focused primarily on teaching and learning. Based on the specific needs of computer science and the conditions of the existing research ecosystem, several personas and use cases were developed in a broad, iterative community process, which were then condensed into a consistent architecture blueprint integrated into the existing landscape. This model is particularly characterized by the preservation of the autonomy of existing operators and services, as well as by a graduated security concept with five protection zones in the sense of zero trust.

The developed architecture serves as a blueprint for the implementation and rollout of the respective services, and also as a means of communication with other providers to establish interoperability with their services. Even though we do not rule out further refinements to the architecture in the course of this process, its basic principles should remain stable. Specific interfaces and data structures still need to be specified and negotiated in detail between the parties involved.

Please note that we followed a technical implementation perspective here. Due to space constraints, we were unable to include other equally important parts of the IT architecture (with regard to business processes, ownership, operation and maintenance, etc.) here, but they are of enormous importance as the starting point for all considerations and for communication processes with relevant stakeholders. Nevertheless, this work provides us with a valuable starting point for further discussion on value streams, business models and governance structures, which is part of our current endeavor.

Additionally, the purpose of this architecture is to give an overview of the services and their interaction with services outside the NFDI. For simplicity, this proposed architecture assumes that NFDIxCS and NFDI services only interact with trusted services. However, we are aware that this will not always be the case in practice. To illustrate how such scenarios can be handled, we outline how a

mechanism can be integrated to manage untrusted services: when a service is not trusted, a representative in the form of a gateway/proxy service is created in the entry zone. This representative acts as the sole point of contact for the untrusted service, mediating and translating its requests, validating and filtering the exchanged data, and limiting the scope of access to only what is strictly necessary. In this way, the untrusted service never interacts directly with the internal infrastructure, ensuring high security.

Our architecture and the services it contains could help to further differentiate HERM business capabilities for the RDSM area. Currently, management of research data is a single capability in HERM, and preserving research software is not yet covered. Although our results are stemming from a national, disciplinary RDSM consortium, empirical grounding should have made the results generalisable and transferable.

Acknowledgements

This work was partially funded by the German Research Foundation (DFG) in the project NFDIxCS under contract number 52/1 – 501930651. We are deeply grateful to the project members for the intense cooperation in the development of this architecture. Special thanks go to Michael Goedicke and Gregor Engels for the productive discussion of software architectures.

References

Ahlemann, F., Stettiner, E., Messerschmidt, M., & Legner, C. (2012). *Strategic Enterprise Architecture Management: Challenges, Best Practices, and Future Developments*. Springer.

Ateia, S. (2025): LLM-Based Information Extraction to Support Scientific Literature Research. Proc. Semantic Web in Libraries Conference (SWIB25). https://doi.org/10.5281/zenodo.17657732

Barker, M., Chue Hong, N.P., et al. (2022). Introducing the FAIR Principles for research software. *Scientific Data*, 9, 622. https://doi.org/10.1038/s41597-022-01710-x

Benzinger, B., Castro Zamora, J.I., Lapuente, J., & Knoth, A (2025). From Theory to Practice: Making Interoperability Become Reality in European University Alliances. *Proc. European University Information Systems (EUNIS)*, EPiC Series in Computing, 107, 98-107. https://doi.org/10.29007/fv2r

Bernoth, J., Ayon, S.I., Al Laban, F., Bavendiek, S., Federrath, H., Striewe, M., Goedicke, M. (2025). Workflow for Creating and Sealing a Research Data Management Container (RDMC). *Proc. Software Engineering (SE 2025) – Companion Proceedings*. https://doi.org/10.18420/se2025-ws-26

Bernoth, J., Al Laban, F., Lucke, U. (2024): Utilizing Personas to Create Infrastructures for Research Data and Software Management. *Proc. INFORMATIK 2024*, 2065-2072. https://doi.org/10.18420/inf2024_180

Blessing, C.B., Khan, S.A., Bernoth, J. (2025). Case Study: Creating a Reusable Execution Environment for WiKoDa. *Proc. Software Engineering (SE 2025) – Companion Proceedings*. https://doi.org/10.18420/se2025-ws-27

CAUDIT (2025): Higher Education Reference Models (HERM), Version 3.2.0. https://www.caudit.edu.au/communities/caudit-higher-education-reference-models/

Chue Hong, N. P., Katz, D. S., et al. (2022). *Fair Principles for Research Software (FAIR4RS Principles).* https://doi.org/10.15497/RDA00068

DFG – Deutsche Forschungsgemeinschaft (2025). Guidelines for Safeguarding Good Research Practice. Code of Conduct. https://doi.org/10.5281/zenodo.14281892

di Cosmo, R. (2020). Archiving and Referencing Source Code with Software Heritage. Proc. 7th Int. Conf. on Mathematical Software (ICMS 2020), LNCS, 362–373. https://doi.org/10.1007/978-3-030-52200-1_36

Druskat, S., Siqueira, A.S., Cushing, R., Lyashevska, O., & Martinez Ortiz, C. (2024) *Citation File Format and cffinit.* Byte-sized RSE, 2(5). https://doi.org/10.5281/zenodo.10838021

Gaia-X – European Association for Data and Cloud (2025). Gaia-X Architecture Document - 25.11. https://docs.gaia-x.eu/technical-committee/architecture-document/25.11/

Goedicke, M., & Lucke, U. (2022): Research Data Management in Computer Science - NFDIxCS Approach. *Proc. INFORMATIK 2022*, 1317-1328. https://doi.org/10.18420/inf2022_112

Ghosh, T. et al. (2021). Zero Trust Core Principles©. White Paper published by The Open Group. https://www.opengroup.org/library/w210

Hartmann, A., von der Heyde, M., Nguyen, D., Zimmermann, H., & Lucke, U. (2025). The Reference Architecture of a National Digital Education Ecosystem. *Proc. European University Information Systems (EUNIS)*, EPiC Series in Computing, 107, 50-59. https://doi.org/10.29007/gjdv

Hasselbring, W., Druskat, S., et al. (2025). Multi-Dimensional Research Software Categorization. *Computing in Science & Engineering*, 27(2), 59 - 68. https://doi.org/10.1109/MCSE.2025.3555023

L'Hours, H., von Stein, I., & Bell, D. (2022). Fair Ecosystem Components. https://doi.org/10.5281/zenodo.6726533

Jones, M.B., Boettiger, et al. (2023). CodeMeta: an exchange schema for software metadata. Version 3.0. https://w3id.org/codemeta/3.0

Lucke, U. (2022). The role of Infrastructure for Software in Open Science. *Proc. Open Science European Conference (OSEC)*, 177-182. https://doi.org/10.4000/books.oep.15829

Manske, A., & Lorenz, A.L. (2024). Base4NFDI – Persona Creation Kit. Zenodo. https://doi.org/10.5281/zenodo.10805470

Marsh, S., & Dibben, M.R. (2003). The Role of Trust in Information Science and Technology. Annual Review of Information Science and Technology (ARIST), 37, 465--498.

Möller, F., Legner, C., Strobel, G., Schoormann, T., Cappiello, C., Loscio, B., & Otto, B. (2025). Data ecosystems in IS research: The road so far, where we are now, and the road ahead. *Electronic Markets* **35**, 111. https://doi.org/10.1007/s12525-025-00860-1

Pohl, K. (2010). Requirements Engineering: Fundamentals, Principles, and Techniques (Vol. 1). Springer Berlin. http://www.springer.com/978-3-642-12577-5

Politze, M., Wieder, P., Schimmler, S., Diepenbroek, M. (2025). Concept for Setting up the Overall Architecture Working Group in the NFDI Section "Common Infrastructures". Zenodo. https://doi.org/10.5281/zenodo.15210894

Rans, J., & Whyte, A. (2017). Using RISE, the Research Infrastructure Self-Evaluation Framework, v.1.1. Edinburgh: Digital Curation Centre. https://www.dcc.ac.uk/guidance/how-guides/RISE

Schmitt, R.H., Anthofer, V., et al. (2020). NFDI4Ing – the National Research Data Infrastructure for Engineering Sciences. *Zenodo*. https://doi.org/10.5281/zenodo.4015201

Schönbächler, M., & Pfister, C. (2017): *IT-Architektur. Grundlagen, Konzepte und Umsetzung*. 2. Ed. Berlin: epubli.

Treloar, A., & Klump, J. (2019): Updating the Data Curation Continuum: Not Just Data, Still Focussed on Curation, More Domain-Oriented. *International Journal of Digital Curation*, 14/1, 87–101. http://dx.doi.org/10.2218/ijdc.v14i1.643

Weill, P., & Ross, J. W. (2010). IT governance. How top performers manage IT decision rights for superior results. [Nachdr.]. Boston, Mass.: Harvard Business School Press.

Williams, M., van de Sanden, M., Scardaci, P. M., Floria, L., & Wierenga, K. (2023). D3.2b - EOSC Architecture and Interoperability Framework. https://eoscfuture.eu/wp-content/uploads/2024/03/EOSC-Future-WP3-GEANT-D3.2b-EOSC-Architecture-and-Interoperability-Framework-2023-06-20.pdf

Wilkinson, M. D., Dumontier, M., et al. (2016). *The FAIR Guiding Principles for scientific data management and stewardship.* Scientific Data, 3(1), 160018. https://doi.org/10.1038/sdata.2016.18

Author biographies

Jan Bernoth is a team leader, researcher and PhD Candidate at the group of Prof. Lucke at the University of Potsdam, Germany. As a Computer Scientist, his research interests lie in infrastructure for Research Data and Software Management, its technical architecture and usability, particularly within the NFDIxCS project. In his dissertation, he examines how researchers can be technically supported in their science communication. As a member of the German Informatics Society, he actively works on Open Science topics in different sections.

Andreas Hartmann is a professor of Applied Computer Science with a focus on distributed applications and their security at HTWK Leipzig, Germany. Prof. Hartmann's research interests lie in the fields of enterprise architecture, IT architecture and IT governance, digital transformation, cloud infrastructures, and IT security. He is the chair of the faculty's examination board and a member of organizations such as The Open Group, EUNIS, ZKI, and CIO e.V. In teaching, he offers courses on web technologies, distributed applications, complex systems, digitization, and IT architecture management.

Ulrike Lucke is a professor of computer science at the University of Potsdam, Germany. Her research activities include institutional infrastructures for education, research and administration. Among other activities, she coordinated a large-scale national initiative to create a digital ecosystem for education across Germany. Until 2018, she was responsible for e-learning and IT strategy as Chief Information Officer of the University of Potsdam. She is a founding member and was vice chair of the German University CIO Association until 2020, and from 2020 to 2024 was a vice president of the German Informatics (GI) association.