Hassan, Mahmoud

Conference Paper — Published Version

BERD Platform: The Journey from User Needs to Agile Development

Suggested Citation: Hassan, Mahmoud (2025) : BERD Platform: The Journey from User Needs to Agile Development, In: Bernoth, Jan et al. (Ed.): deRSE24 - Selected Contributions of the 4th Conference forResearch Software Engineering in Germany, Berlin Universitites Publishing, Berlin, pp. 1-17, https://doi.org/10.14279/eceasst.v83.2629

This Version is available at:
http://hdl.handle.net/11108/661

Electronic Communications of the EASST
Volume 83 Year 2025

deRSE24 - Selected Contributions of the 4th Conference for Research Software Engineering in Germany

Edited by: Jan Bernoth, Florian Goth, Anna-Lena Lamprecht and Jan Linxweiler

BERD Platform: The Journey from User Needs to Agile Development

Mahmoud Hassan

DOI: 10.14279/eceasst.v83.2629

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

BERD Platform: The Journey from User Needs to Agile Development

Mahmoud Hassan [1]

[1] m.hassan@zbw.eu
ZBW – Leibniz Information Centre for Economics, Germany

Abstract: With the movement of the Open Science initiatives, a research data infrastructure where an integrated management of (un)structured data and associated scientific software in science, economics and business is highly desired by researchers, libraries and institutes which is inline with open software and standards and aligned with the FAIR principles (Findable, Accessible, Interoperable, and Reusable). The BERD@NFDI consortium aims to establish a research data management platform for Business, Economics, and Related Data (BERD) within the German National Research Data Infrastructure (NFDI). The platform will host diverse resources, such as research data from areas like marketing, analytics, and social sciences, involving various partner institutions and user communities. This results in an agile, user-driven requirements engineering process. In this paper, I present the steps we took in our Software Development Team (SDT) from gathering community requirements through formalizing these requirements to the agile software engineering process of building the BERD platform. An evaluation of the different evolving measures and practices taken for the development of the platform will be discussed.

Keywords: RSE practice, Agile development, InvenioRDM

1 Introduction

Developing a research data management platform is a complex process that requires various technical perspectives to be considered while gathering the requirements of the users and the targeted communities. A subset of these technical perspectives includes sharing research data and results that abide by the FAIR principles [WDA+16] and ensuring a sustainable preservation and curation process, which guarantees that high quality data is verified and research data integrity is achieved [AW22]. One of the goals of the BERD@NFDI consortium[1] is to build a powerful research data management platform for collecting, processing, analyzing, and preserving Business, Economic, and Related Data. It is designed as a comprehensive platform for managing and sharing research data in the business and economics fields in Germany. Its main purpose is to streamline the management, sharing, and reuse of research data, with a focus on unstructured data for the BERD communities which consists of domain-driven researchers in business, economics, and other social sciences, and NLP practitioners and data scientists. The platform aims to integrate with the broader NFDI (National Research Data Infrastructure) initiative, ensuring

[1] BERD@NFDI is funded by the German Research Foundation (DFG) within the framework of the NFDI – project number: 460037581

interoperability, scalability, and compliance with national and international standards for data management. In this paper, the processes and approaches that the Software Development Team (SDT) performed to realize the BERD platform will be discussed. First, the requirement gathering process, how use cases were represented, and the multi-step visualization techniques will be illustrated. Then, details on the ideas and approaches that were developed to extend and reuse an open-source research data management software will be discussed. An overview of how the development work was deployed and the cloud-based infrastructure that was used will also be mentioned. Finally, an evaluation of the approaches that were taken in the development will be assessed.

2 Use Case representation and Multi-step visualization

The BERD@NFDI consortium consists of several partner institutions and user communities from areas such as marketing, analytics, and social sciences, where their goals vary in terms of fulfilling of their user needs. Some of these goals overlap, for example, in the ability of users to browse and search research data, taking into consideration the different search filters for each resource type provided by the community. In addition to the community requirements, there are also technical requirements on how the community requirements can be developed to ensure the integrity of the services and the stability of the platform. Therefore, collaboration and clear communication of requirements among all key stakeholders are necessary to ensure successful research data management [CM15]. The process starts with formalizing the community requirements and translating these requirements into visual interfaces to reflect on them and integrate them later into a prototype.

2.1 Use Case Representation

The first step undertaken was to gather the diverse requirements of the users and demonstrate the priorities of these use cases. The use cases have been represented as user stories. A user story is a demonstration of a goal that the intended user should achieve when they use the targeted software [AM12]. It consists of the goal, the intended actors (users), a pre-condition that needs to exist for the user to pursue an action, a trigger to activate the user's motive to perform an action, a basic flow or the main success scenario of how the user pursues the goal, and alternative flows, if they exist, on it could be pursued. User stories are an essential building block at the start of a software engineering process to facilitate and ensure transparent alignment of ideas between the community requirements and the technical requirements [GD20]. The key user requirements are:

1. Data Collection and Management
   - Requirement: The platform must support the collection, curation, and management of a wide range of data types, specially on unstructured data.
   - Complexity: High. This involves developing robust data ingestion pipelines, handling various data formats (e.g., CSV, JSON, XML), and ensuring data integrity and quality. Implementing metadata standards and ensuring compliance with FAIR

(Findable, Accessible, Interoperable, and Reusable) principles adds another layer of complexity.

2. Data Access and Sharing
   - Requirement: Users need seamless access to the platform's data, with the ability to share datasets within the research community and beyond, while respecting privacy and confidentiality agreements.
   - Complexity: High. This requires implementing fine-grained access control mechanisms, ensuring GDPR compliance, and possibly integrating with authentication systems like ORCID. The platform must also support various data-sharing modalities, from public data releases to controlled access environments for sensitive data.
3. Interoperability and Integration
   - Requirement: The platform must be interoperable with other NFDI services and external data platforms, supporting data exchange and integration.
   - Complexity: Very High. Achieving this requires adherence to interoperability standards (e.g., OAI-PMH for metadata harvesting, RESTful APIs for data exchange) and developing connectors or adapters for various external systems. Ensuring seamless integration while maintaining data integrity and performance is challenging.
4. User Interface and Experience
   - Requirement: The platform should offer an intuitive, user-friendly interface that serves to a wide range of users, from data scientists to domain-specific researchers.
   - Complexity: Medium. Designing a user interface that balances ease of use with powerful functionality requires careful planning. It must accommodate users with different levels of technical expertise while providing a seamless experience across devices. The complexity is compounded by the need to ensure accessibility (e.g., WCAG compliance) and internationalization.
5. Scalability and Performance
   - Requirement: The platform must be scalable to handle increasing amounts of data and concurrent users, while maintaining performance.
   - Complexity: Very High. Ensuring scalability requires robust architectural planning, possibly involving cloud-native technologies (e.g. microservices, container orchestrators) and distributed storage solutions. Performance optimization for both data processing and user interactions, especially under heavy load, is a significant challenge.
6. Data Preservation and Archiving
   - Requirement: The platform must provide long-term data preservation and archiving solutions, ensuring that data remains accessible and usable over time.

- Complexity: High. Implementing data archiving requires adherence to standards like OAIS (Open Archival Information System) and ensuring that data formats are preserved or migrated to prevent obsolescence. This also involves creating and maintaining persistent identifiers (e.g., DOIs) for datasets.

The documentation and illustration of the user requirements and stories were collaborative work between the partner institutions in the BERD@NFDI consortium and the SDT.

2.2 Wireframes

After reaching an agreement and documenting the various ideas of the BERD community, working on a prototype was initiated. Prototyping has been identified as essential for the incremental improvement of the utility (i.e., usefulness) and usability (i.e., ease of use) of an application, especially in Usability Engineering, where continuous feedback from users is acquired for continuous enhancement and improvement of the prototype [Nie92]. A visualization of the user stories has been developed to confirm on how they are reflected and to reach an initial agreement on how the community's ideas are to be demonstrated. The SDT began a two-step process by first creating the wireframes and then creating the mockups. A wireframe is a rough visual outline of a proposed application [Llo09]. It is an abstract view of how a user story could be visualized. It is a specific kind of prototype used in the development process and generated per user case owner during the user-centered design process to reflect the user stories and receive feedback from use case owners on the accuracy and precision of the ideas conveyed. So far, the creation of the wireframes has been based on the user stories of the different use case owners. These wireframes were created individually with the partner institutions and interlinked with a site map, as shown in Figure 1.

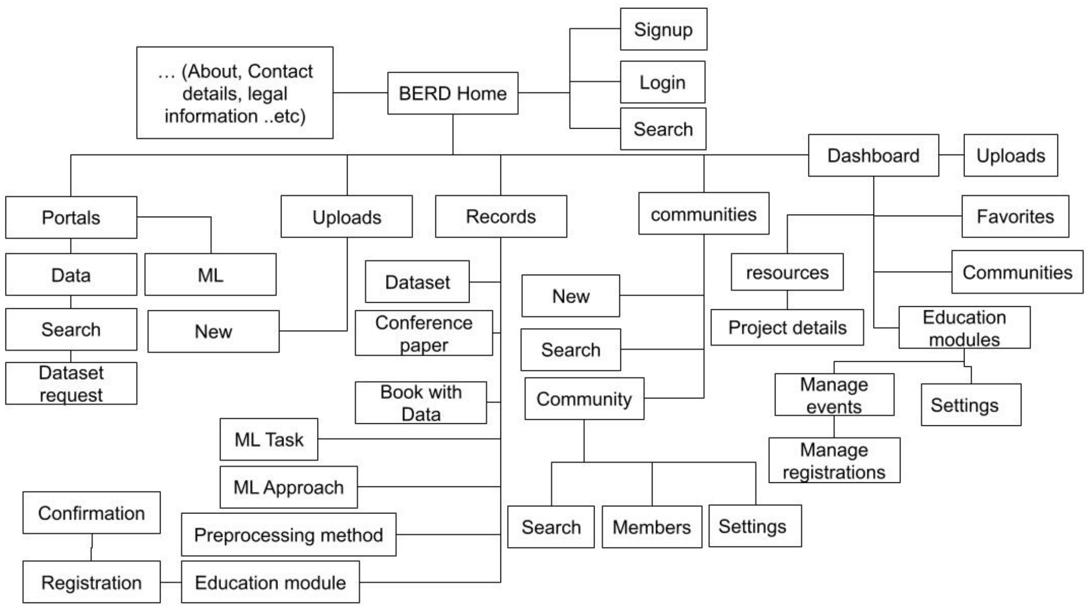

Figure 1: BERD site map

The figure is illustrating an example of a primitive site map showing the components that was discussed to sketch the user requirements. It clarified the points of intersections and overlaps inaddition to distinct entities. For example, the BERD home is the main component of the BERD platform which from it derives the different actions and workflows for example to login, sign up, upload research data or to view the user dashboard and what actions the user could take whether to check their datasets that they bookmarked as favorite or to check the communities that they joined or education modules that they participated in. The figure also collected the types of research data derived from the use cases which would be needed to create as different components and representations of data which could be for example a dataset, or a conference paper , a machine learning (ML) task or education module ..etc . It also illustrated that some research data were not only for viewing or downloading but further access actions may needed to be performed for instance for the Education module, it would require to go through a registration workflow and confirmation to access it's material. It is important to mention that some of the illustrated components in the site map were taken out or merged into other components or are still under development.

2.3 Mock Ups

The next step of prototype development, after creating all the wireframes from the different user stories, was to create the mock-ups. The mock-ups in this stage of development are an interactive non-functional interfaces that integrates the different wireframe concepts in the site map shown in Figure 1. They give the user the chance to navigate the site map and try out an active, more realistic, and integrated interface of the different BERD community user stories. This allows for testing targeted user interactions and gathering feedback on possible improvements in terms of design and how well the user stories are integrated, grouped, and interpreted to identify any overlap and divergence of the various ideas from the different user community backgrounds. Early and continuous user involvement has revealed not only the ambiguity of the user needs but also highlighted the user values [Kuj08]. The mockups were deployed to a server, as shown in Figure 5. More details on the infrastructure used by the SDT to carry out the development work will be briefly specified in Subsection 4.1. Figure 2 shows an example of a partial illustration of the BERD home page as part of the mockups that were developed which shows the main components and actions that the user may take when accessing the platform. Feedback sessions were offered to the use case owners in an agile approach with iterative sprints, and continuous improvement, and deployment, which were later used in the development of the BERD platform. The page is not functional, for example the searching functionality in the figure will not execute a search query but it would navigate to the search results component to illustrate how the workflow and the scope of searching for the research data would happen, which depends on where the user execute the search functionality, for example whether it happen in the home page or in one of the portal pages.

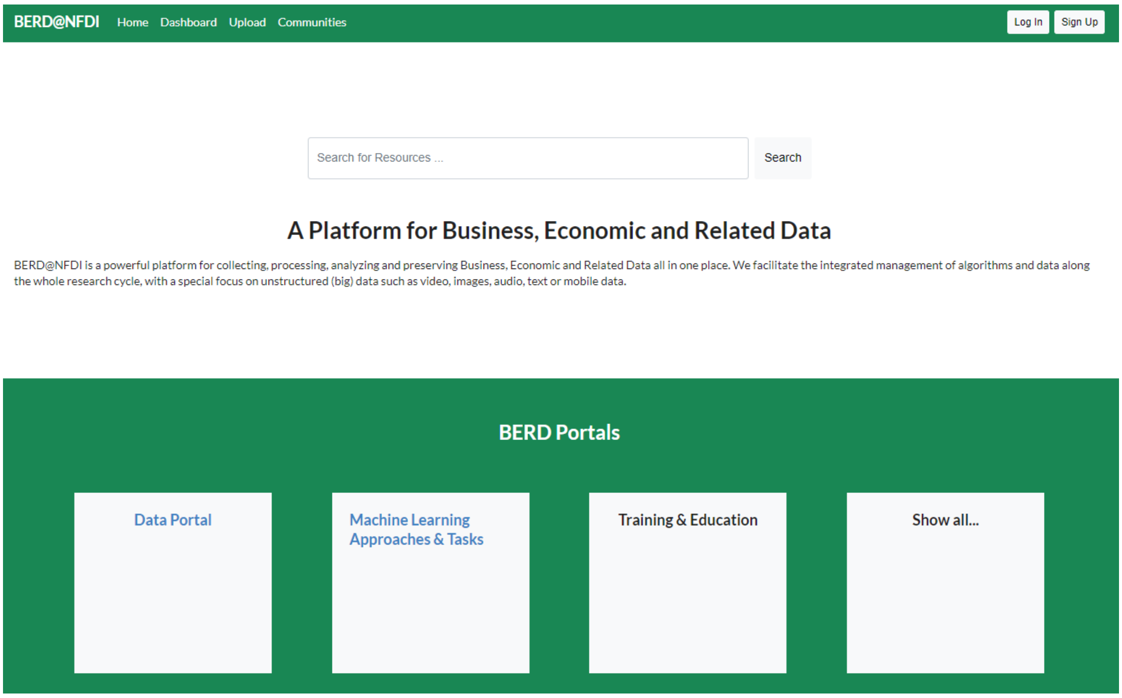

Figure 2: BERD mockups

3 Agile Platform Development

The development of the functional platform was ready to begin after gaining a clear understanding of the use-case owners' ideas and values, and after completing the visualization and confirmation phases through wireframe creation, mockup navigation, and feedback sessions. An adaptable and flexible approach was adopted by developing the platform in sprints and using iterative approaches for each use-case owner, aligning as much as possible with the Agile methodology [HC01]. Prioritizing the features to be implemented with a focus on effectiveness, constant deployment to receive rapid feedback, and maintaining an open loop of communication with the use-case owners formed the development cycle used by the SDT during the development of the BERD platform.

3.1 Developing BERD with InvenioRDM: Compliance and Standards

The BERD platform is developed on top of the open-source research data management application InvenioRDM. InvenioRDM is a turnkey research data management (RDM) repository platform based on the Invenio Framework and Zenodo [Cc24]. InvenioRDM is being chosen in compliance with the consortium's support of the Open Science paradigm. The development phases of InvenioRDM is in compliance with the Community Framework for Good Practices in Repositories (COAR) [GCV+21]. COAR [oOAR22] is an international initiative aimed at promoting best practices for managing and developing repositories that store and share research outputs. It provides guidelines and recommendations to ensure that repositories are interoperable, sustainable, and aligned with global standards. It emphasizes open access, metadata quality, and the integration of repositories within the broader research ecosystem. By adhering to COAR's framework, repositories can ensure compliance with emerging technologies and policies, supporting the global advancement of open science. This accordingly promotes the research output to be FAIR-compliant which are essential for making data discoverable and reusable. The choice of invenioRDM as well offers several benefits, including access to a large community of Invenio users and developers, which provides robust technical support. Additionally, Invenio facilitates the integration of key standards, such as Digital Object Identifier (DOI) for persistent identifiers and DataCite for metadata schemas. An overview of the functional architecture structure of the BERD platform in its current state is shown in Figure 3. The Figure shows the main functional components of the BERD platform that has been developed so far. Some functional components like the Login, Sign up, Administration has been taken over and used via InvenioRDM without further modifications other than thematic and adding GDPR compliance in the authorization workflow. The Upload and Search component have been highly modified to include the user requirements which included high modifications into the modules that InvenioRDM host their services. The Data portal and Services portal has been implemented by developing the BERD module which made use of the services that is offered through InvenioRDM. OpenSearch is used to power the BERD's search and indexing capabilities. It enables fast, full-text search across metadata records and content, ensuring that users can efficiently retrieve and filter research data and other related data. OpenSearch handles complex queries, facets, and relevance ranking to provide an optimized search experience. PostgreSQL is used as the primary relational database to store and manage structured data, such as metadata records, user information, and access permissions. It ensures data integrity and supports complex relationships between entities like datasets and users. In the following subsections, more details of the ideas and approaches on how InvenioRDM was extended and reused to fulfill user needs and requirements in developing the BERD platform will be discussed.

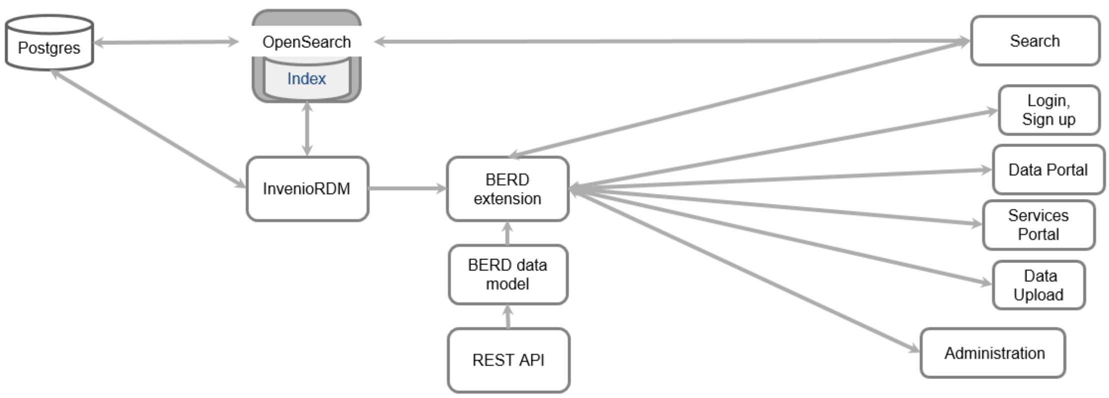

Figure 3: Functional architecture of BERD platform

3.2 Extension of InvenioRDM

InvenioRDM is characterized by its modular architecture, consisting of several modules organized into distinct layers encompassing data access, service, and presentation, which renders the code structure easily comprehensible. This layered architecture also facilitates the construction of custom modules, allowing seamless extension according to the specific functionalities desired by the use-case owners. For instance, it enables the addition of new resource types of research data, implementation of fine-grained search capabilities, and execution of quality checks for the data presented on the platform. Figure 4 depicts a brief abstract view of the modular architecture of InvenioRDM on which the BERD platform is built. For example, invenio-administration[2] is an Invenio module that provides administrative functionalities for the BERD platform. As of the current beta version v12, which the BERD platform has updated to in March 2024, this module offers an administration panel for admin users to verify, block, and delete users, as well as delete records, grant moderation capabilities, and specify valid user domains for platform registration, among other admin actions. Another example is the invenio-rdm-records module[3], which has recently integrated the react-invenio-deposit module[4], a modular library for building user interfaces for research data deposit workflows. InvenioRDM offers the flexibility to enable or disable completely or parts of its features via a configuration file that is provided and overwritten by the value of the configuration variable for the related module. It is also possible to extend InvenioRDM with customized modules, as was done in the BERD platform by integrating the BERD module, which contains the customized workflows and features required by the BERD community. Examples include the Services Portal and the Data Portal, as shown in Figure 3.

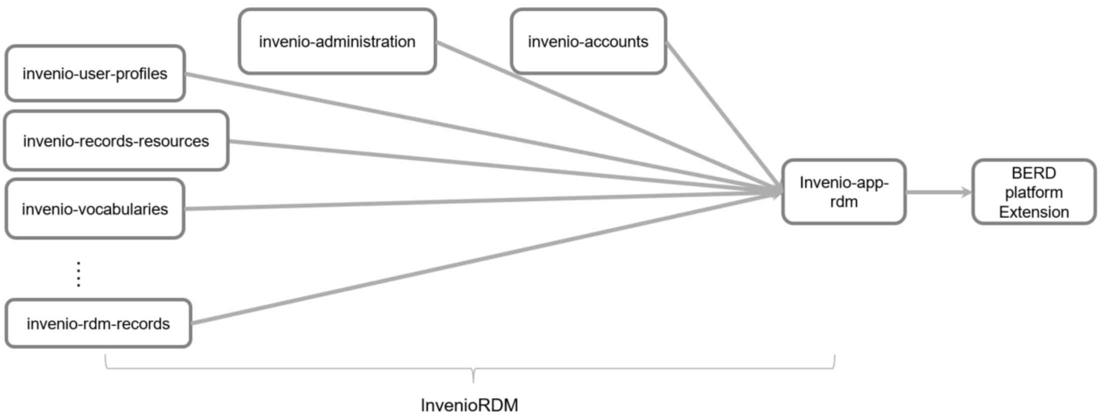

Figure 4: Modular architecture of BERD

3.3 Communities as quality gates

A community is an organizational feature in InvenioRDM used to group a related collection of research outputs and data based on a specific theme, such as a project, department, conference, or

[2] Reference of the module is given at https://github.com/inveniosoftware/invenio-administration
[3] Reference of the invenio-rdm-records module is given at https://github.com/inveniosoftware/invenio-rdm-records
[4] Reference of the react-invenio-deposit module is given at https://github.com/inveniosoftware/react-invenio-deposit

institution. Originally in InvenioRDM, logged-in users could create a community with their own purpose or theme and invite members with related interests to gather data relevant to the community's purpose. In the BERD platform, however, communities are not visibly promoted and are primarily reused as quality gates for the data uploaded to the platform. Several communities have been created corresponding to the resource types currently existing on the platform. Members of the use-case owners who requested these resource types have been assigned as community managers to review the data uploaded to the platform, ensuring that high-quality and valid data are published. To achieve this workflow, multiple aspects needed to be implemented:

1. Community metadata with specific descriptions and access policies needed to be predefined.

2. Encrypted user data needed to be added and injected during the deployment stage to assign community managers.

3. The deposit form behavior needed to be altered so that no research data is published directly to the platform; instead, it is forwarded first to the respective communities to be reviewed before publishing.

3.4 Promoting Datasets

One of the goals of BERD@NFDI is to provide high-quality data on the platform. One strategy to attract researchers is to feature a 'Dataset Of The Month' from the Data portal on a monthly basis. This feature is a requirement from a use case owner, the University of Mannheim, which is a partner university in the BERD@NFDI consortium. The platform utilizes services from InvenioRDM and implements them in the BERD module to retrieve the promoted record ID and gather relevant information such as the title, description, and related publications that cites the datasets, encouraging users to access and download them fully. This feature includes multiple steps to achieve:

1. The featured record id gets retrieved using a proxy service from an invenio module, invenio-rdm-records.

2. A customized implemented service for retrieving related publications of the specified record is reused to retrieve the necessary information for the featured record.

3. An alternative flow and an exception handling is executed in case the record or the required related information does not exist or is restricted.

4. A customized view is implemented to represented the interface of the feature and inject it in the BERD module.

5. A customized configuration variable is added to the configuration file of the BERD extension module which is adjusted on a monthly basis.

3.5 Representation of Research Data

With the different values that comes with the requirements that the BERD community has to their intended users, different resource types of research data with distinct representation and significance of the details of the research data is applied. The platform categorizies research data based on their resource type and assigns the appropriate metadata elements accordingly. This customization involves overwriting and modifying the templates within the invenio-app-rdm module[5] inside the BERD extension module, which are Jinja templates[6]. Conditional structuring of sections of the landing page of the research data records in the BERD-module together with refactoring the research data record templates enabled the right representation of research data information tailered to the needs of the use-case owners.

[5] Invenio App RDM module is the main module that represents the repository application, the module is found at https://github.com/inveniosoftware/invenio-app-rdm

[6] Jinja is a web template engine for the Python programming language which are used across the different invenio modules, the documentation is given at https://jinja.palletsprojects.com/en/3.1.x/

4 Infrastructure and Deployments

An overview of the infrastructure that hosts the BERD development deployments and the workflow of the deployments and how the deployments are done is briefly discussed.

4.1 Infrastructure

The SDT uses a cloud-based infrastructure to host the deployments that were developed throughout the development phases as shown in Figure 5. IONOS[7] is the cloud provider that hosts the BERD infrastructure. The infrastructure is setup using IONOS's Data Center Designer which provides a GUI for setting up servers and other components . IONOS is used to host the deployments where it consists of the mockup server, the staging servers, and the production servers that encompass five dedicated core servers. The mockups consist of one server where it hosts the Docker-LAMP stack[8]. Each the staging and the production consists of two servers, one server hosts the required services (i.e redis, rabbitmq, open search) and the other server hosts the application (i.e UI components, API components, Celery workers .. etc). The staging and production servers have been created following InvenioRDM's recommended architecture[9].

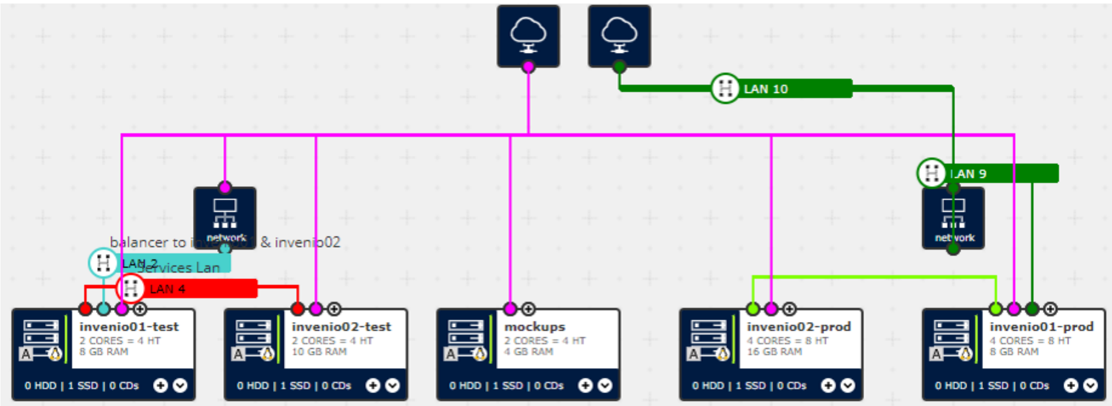

Figure 5: BERD infrastructure showing the staging , mockups and production servers

4.2 Deployments

The deployments of the BERD platform has been implemented via the Gitlab Continuous Integration/Continuous Delivery (CI/CD) pipeline. An overview of how the pipeline works can be seen in Figure 6. The illustrated process applies to the staging and production servers and not to the mockups server. After the development work of a feature in a specific branch has been implemented by the developer who is assigned the task and reviewed and approved by a different

[7] IONOS is a German Internet service provider that became known primarily for its web hosting, domain and cloud computing products.

[8] Docker-LAMP is a set of docker images that include the phusion baseimage (18.04 and 20.04 varieties), along with a LAMP stack (Apache, MySQL and PHP) all in one handy package. Usage and documentation is given at https://github.com/mattrayner/docker-lamp

[9] Documentation is given at https://inveniordm.docs.cern.ch/develop/architecture/infrastructure/

developer in the SDT, the feature branch is merged to the main branch. This action triggers the deployment of the latest code base with the new merged branch to the staging server. The deployments goes to three stages that needs to be passed sequentially. The first stage is the testing stage which consists of a number of testing jobs that tests different aspects in the BERD platform to make sure for instance that the new feature does not break or disturb the functionalities that already exists in the platform, new tests could be added to this stage in an iterative pace based on new developments. The next stage is the build stage, which for example builds the working directory and installs all the required dependencies for the BERD application that Invenio needs. Lastly, there is the deployment stage, where two scenarios appear:

1. The deployments are executed automatically and gets deployed to the staging server. The staging server is then used for testing before deploying the latest changes to the production server.
2. To deploy the changes that are merged to the main branch on the production server, creating a tag in Gitlab[10] and protecting it, triggered the deployment action to the production server.

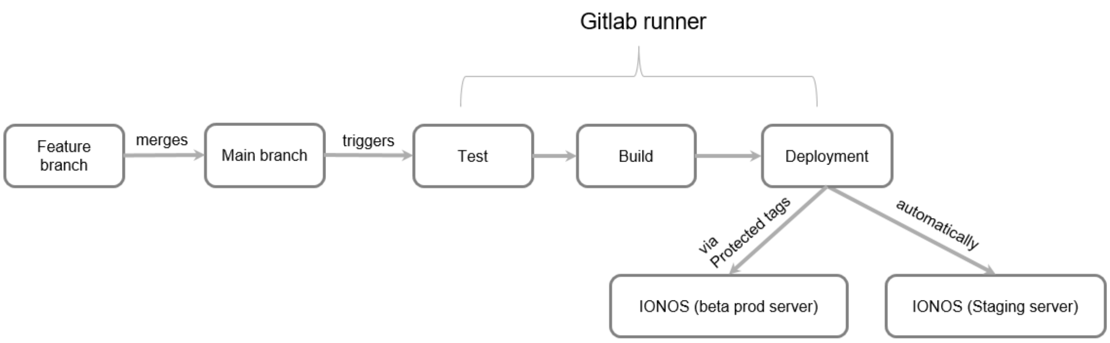

Figure 6: Deployments of BERD platform

5 Evaluation

5.1 Re-usability vs Extensibility

During the development of the platform requirements, discussions often arose on how to achieve certain requirements. There are multiple approaches to implement a certain requirement depending on the use-case to be developed. A subset of these approaches is as follows:

1. Modify the configuration variables of InvenioRDM modules, which was assessed as the simplest approach.
2. Modify or overwrite the existing templates in a certain Invenio module.

[10] For full usage of tags is given at the Gitlab documentation https://docs.gitlab.com/ee/user/project/repository/tags/

3. Implement custom views and/or extend the views with React components.

4. Fork an entire module and partially refactor it to sync with the requirements.

5. Reuse chunks of code from an Invenio module into the BERD module, which would lead to more flexibility with high maintenance debt and occasional synchronization with the module updates.

An example of reusing of an Invenio module is discussed. Due to a use-case requirement to provide a systematic workflow for users to upload their research data in a sequential flow in the deposit form, which is different from the implemented deposit form in Invenio, in addition to the different specific metadata requirements that should show depending on the resource type of the research data, an approach on how to develop such a requirement led to evaluating whether to fork of a module or partially refactor it and include it in the BERD extension. Due to the continuous requirements of shaping how users upload their data, this led to the forking of an Invenio module, which is the react-invenio-deposit module. An abstract representation of the reusable/fork approach on how the fork was made is shown in Figure 7. In the figure, the Invenio module was forked and reused to apply BERD use requirements to fulfill the use-case of a customized deposit form. By then, the BERD platform was running on version 10 of InvenioRDM. After an upgrade to InvenioRDM beta version 12 was performed, the react-invenio-deposit module was moved inside of an invenio module as shown in Figure 8, namely invenio-rdm-records. Then it was necessary to revisit the approach applied in Figure 7 to extend a partial of the code that was reused in the react-invenio-deposit module and include it into the BERD-module extension. In this case extending the invenio module was more sufficient than forking the whole module where high customization was possible without impacting other Invenio modules. This reduced the need to maintain a whole Invenio module, synchronize it with the upstream module, or resolve the conflicts that appear occasionally.

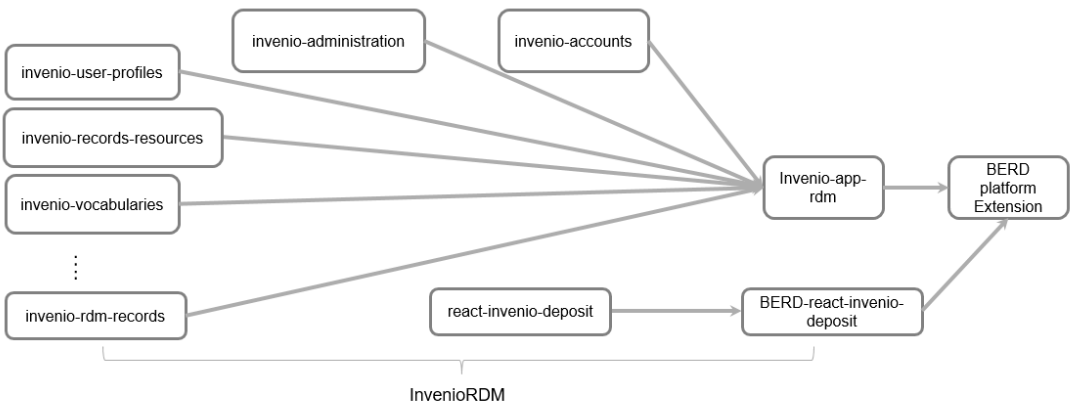

Figure 7: Reusable approach

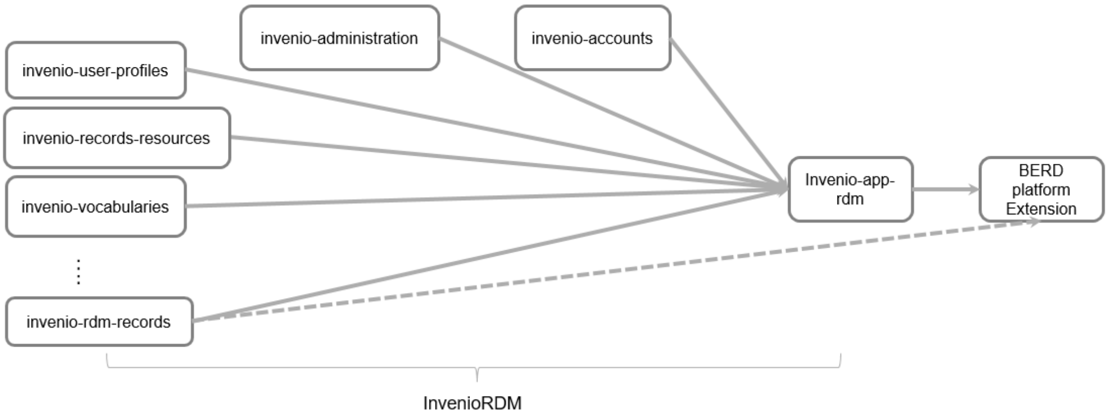

Figure 8: Extensible approach

5.2 Deployments Mechanism

Another approach was used for deployments different from what was explained in Subsection 4.2. The approach was that there were two branches named develop and main. And develop branch was used to be the latest code changes where once merging to it, the latest changes got deployed to the staging server. And once it was decided to deploy the latest changes to the production server, the develop branch was merged into the main branch and the latest changes get deployed to production. This approach imposes risk of mistakenly merging from the feature branch to the main branch directly which would cause merge conflicts or cause production issues. The deployment mechanism got updated by what was explained in Subsection 4.2 to avoid such issues and to provide an extra layer of assurance when the SDT proceed with the action of deployment.

5.3 Maintainability

Despite the high modularity of extending and reusing InvenioRDM modules, it was observed that a high level of adjustment and refactoring of core modules like the react-invenio-deposit which is, as explained, responsible of rendering the deposit forms, would lead to high maintenance efforts in the technical side. It depends also on the refactored components and to what extent was the module refactored, which would lead to technical dependencies that need to be adjusted accordingly, especially with an upgrade to newer versions of InvenioRDM. Otherwise, the various invenio modules are extensively tested for its services and functionalities and has an active community of open collaboration which keeps track of arising issues and a quick response to bug fixes and dependencies conflicts.

6 Conclusion

The BERD Platform has been launched and presented in its beta version at the beginning of March 2024. It is successfully engaging its target audience of researchers in the economic and social sciences. The breadth of participation, positive feedback, and diverse set of tools and datasets point to a promising future for the platform in transforming research data management in these fields. The continuous feedback from targeted groups that have been accompanied through the process of the development of the platform supported into achieving this outcome. There has been high demand to register on the platform and users uploading their research data which the data stewards review before publishing on the platform. Additional feedback and comments have been received from users after launching to improve certain aspects such as the usability. The platform is also currently using test DOIs in its beta state which effects the Find-ability aspect and work is still needed as well on improving its FAIRness especially in the accessibility and interoperability aspects [SLZ+23] once persistent identifiers are setup which is one of the key requirements for integration with external platforms, data exchange and long term preservation. The BERD platform is still rapidly evolving and is constantly under development to reach to a stable productive environment and to improve its functionalities and include more services for the users.

7 Author contribution and Acknowledgments

I extend my gratitude to David Glück and Franz Osorio for their valuable contributions to various phases of the BERD platform's development as former members of the Software Development Team (SDT) at ZBW – Leibniz Information Centre for Economics. I also thank Esteban Gabancho, a freelance InvenioRDM developer affiliated with the University of Mannheim, for his support in deploying the BERD platform and providing valuable consulting during its development.

The publication practices at my affiliated institute adhere to the CRediT guidelines [11], and therefore the contributions of the team members are acknowledged as follows:

- Writing – Original Draft Preparation : Mahmoud Hassan.
- Conceptualization - Software - Methodology - Validation : Mahmoud Hassan - David Glück - Franz Osorio.
- Supervision: Klaus Tochtermann.
- Funding Acquisition: Klaus Tochtermann, Atif Latif, Fidan Limani.
- Data Curation: Atif Latif, Fidan Limani.
- Project Administration: Anja Busch, Ahmed Saleh.

8 Funding

BERD@NFDI is funded by the German Research Foundation (DFG) within the framework of the NFDI – project number: 460037581

[11] Contributor Roles Taxonomy, reference is given at :- https://authorservices.wiley.com/author-resources/Journal-Authors/open-access/credit.html

Bibliography

[AM12] V. K. Anu Kankainen, Kirsikka Vaajakallio, T. Mattelmäki. Storytelling Group – a co-design method for service design. *Behaviour & Information Technology* 31(3):221–230, 2012.
doi:10.1080/0144929X.2011.563794

[AW22] J. R. Angeliki Andrikopoulou, G. Walton. Research Data Management (RDM) and the Evolving Identity of Academic Libraries and Librarians: A Literature Review. *New Review of Academic Librarianship* 28(4):349–365, 2022.
doi:10.1080/13614533.2021.1964549

[Cc24] CERN, contributors. InvenioRDM: the turn-key research data management repository. 2024.
https://invenio-software.org/products/rdm/

[CM15] E. Chiware, Z. Mathe. Academic libraries' role in Research Data Management Services: a South African perspective. *South African Journal of Library and Information Science* 81:1–10, 12 2015.
doi:10.7553/81-2-1563

[GCV+21] S. Gonzales, M. B. Carson, G. Viger, L. O'Keefe, N. B. Allen, J. P. Ferrie, K. Holmes. User Testing with Microinteractions: Enhancing a Next-Generation Repository. *Information Technology and Libraries* 40(1), Mar. 2021.
doi:10.6017/ital.v40i1.12341
https://ital.corejournals.org/index.php/ital/article/view/12341

[GD20] A. C. C. B. Giacomo Poderi, Hanna Hasselqvist, V. D'Andrea. Matters of concerns and user stories: ontological and methodological considerations for collaborative design processes. *CoDesign* 16(3):220–232, 2020.
doi:10.1080/15710882.2018.1557694

[HC01] J. Highsmith, A. Cockburn. Agile software development: the business of innovation. *Computer* 34(9):120–127, 2001.
doi:10.1109/2.947100

[Kuj08] S. Kujala1. Effective user involvement in product development by improving the analysis of user needs. *Behaviour & Information Technology* 27(6):457–473, 2008.
doi:10.1080/01449290601111051

[Llo09] D. Lloyd. Evaluating human-centered approaches for geovisualization. September 2009. Unpublished.
https://openaccess.city.ac.uk/id/eprint/468/

[Nie92] J. Nielsen. The usability engineering life cycle. *Computer* 25(3):12–22, 1992.
doi:10.1109/2.121503

[oOAR22] C. of Open Access Repositories. COAR Community Framework for Good Practices in Repositories, Version 2. Sept. 2022.
doi:10.5281/zenodo.7108101
https://doi.org/10.5281/zenodo.7108101

[SLZ+23] J. Saldanha Bach, F. Limani, Y. Zhang, A. Latif, B. Mathiak, P. Mutschke. FAIR Assessment Practices: Experiences From KonsortSWD and BERD@NFDI. *Proceedings of the Conference on Research Data Infrastructure* 1, 2023.
doi:10.52825/cordi.v1i.344
https://www.tib-op.org/ojs/index.php/CoRDI/article/view/344

[WDA+16] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3(1):160018, Mar 2016.
doi:10.1038/sdata.2016.18