Setting up a Research Data Repository Based on Invenio RDM: An Experience Report

Herbert Lange[1,*]

[1]Språkbanken Text, Department of Swedish, Multilingualism, Language Technology, University of Gothenburg, Sweden

Abstract

Setting up a research data repository is a time-consuming task. Learning from others who went a similar way can be very helpful. For that reason we want to share our own experience. This article is based on the lessons learned while setting up a repository based on Invenio RDM at the Leibniz Institute for the German Language (IDS) in Mannheim between March 2022 and September 2023. We explain our decisions and steps taken on the way. Even though our requirements differ from other institutions' we are confident that this description can be helpful to others. We mostly worked with language data, still the basic considerations are as valid and relevant to all areas within the Digital Humanities and beyond. This article is neither intended to convince you to set up your own repository nor to do the opposite. It should, however, help you to make your own informed decisions.

Keywords

Research infrastructure, Research data management, Digital preservation, Invenio RDM

1. Introduction

Every research institution has to face the question where to store the data that has been produced by its affiliated researchers. At the same time researchers themselves are confronted with stronger requirements by funding and state agencies on how and how long their data has to be stored (e.g. Swedish Research Council 2022, p. 35 and European Research Council Scientific Council 2022, pp. 4). One solution for the institution in question could be to host their own research data repository. Depending on the available resources, both monetary and human, this can be a very reasonable decision. Otherwise it is usually possible to store research data either in public repositories such as Zenodo[1] or within a common research infrastructure.[2].

In the Section 2.1 we will present some considerations about hosting a repository, including the criteria for the repository at the Leibniz Institute for the German Language (IDS). In Section 2.2 and 2.3 we will showcase the resulting system. We will conclude with a broader discussion in Section 3.

DNHB 2024: Digital Humanities in the Nodic and Baltic Countries 8th Conference, Reykjavík, Iceland, 27–31 May 2024.

*Corresponding author.

herbert.lange@svenska.gu.se (H. Lange)

0000-0002-1450-5486 (H. Lange)

Digital Humanities in the Nordic and Baltic Countries Publications – ISSN: 2704-1441

[1]https://www.zenodo.org (CERN and OpenAIRE 2013)

[2]Relevant infrastructures are CLARIN (https://www.clarin.eu/) for language data or DARIAH (https://www.dariah.eu/) for Digital Humanities in general

2. Setting up a research data repository

Let's assume you want or need to set up your own repository and cannot rely on any external research data repository to host your data. One major decision that has to be taken is the technology on which you want to base the repository. This could be a either a new system completely developed from scratch or a setup based on a range of preexisting software. A third option would be to use a commercial hosted provider. Each of these choices has its advantages and challenges.

A bespoke system has the major advantage that it can exactly match all your requirements. Its drawback is that it requires the means and skilled personnel to develop it. Usually it also takes more time and knowledge to develop a system completely from scratch. One successful example of an in-house development is ARCHE (A Resource Centre for the HumanitiEs)[3] at the Austrian Academy of Sciences (Żółtak, Trognitz, and Ďurčo 2022).

Basing your repository on existing free software has the advantage that you already get many features for free. Candidates for free repository software are Fedora Commons[4], Dspace[5], Invenio RDM[6] and many more. Each of these systems has a set of supported features. The main advantage and a major challenge of using an existing software is that they are usually surrounded by a community. This community takes decisions about future developments and drives them forward. Getting involved in such a community means that you can influence the direction of future development. At the same time being involved in the community also means you should actively contribute towards the development of the software. This of course requires resources: time, money and skills.

Commercial hosted service providers include Preservica[7] and Digital Commons[8]. The main advantage of such a solution is that it requires less personnel and expertise but at the same time has a significant subscription cost. Many of the points discussed here are also relevant for an external solution. However, we focus on self-hosting in this article.

Which of the three solutions to choose has to be based on further considerations and circumstances. Before starting the endeavor of setting up a research data repository, listing requirements should be a first step. In the next section we will present the questions we discussed at length and which can form the foundation for suitable system requirements.

After discussing such questions and formulating requirements you can check out available systems and compare their features. Developing your own system makes it potentially easier to meet all your wishes but has the major disadvantage of additional cost. However, you should be aware that adapting existing software to your needs usually also requires time and skills. In any case, the aspect of future maintenance should not be ignored.

[3] https://arche.acdh.oeaw.ac.at/browser/about-service
[4] https://github.com/fcrepo/fcrepo
[5] https://dspace.lyrasis.org/
[6] https://inveniosoftware.org/products/rdm/
[7] https://preservica.com/
[8] https://bepress.com/products/digital-commons/

2.1. Our Criteria

The IDS has already been providing a research data repository based on Fedora Commons using a legacy version. This repository was certified by the CoreTrustSeal (CTS)[9] and is part of the CLARIN infrastructure making the IDS a CLARIN B centre[10]. This repository had to be replaced by a new one at some point due to increasing maintenance issues. The idea of updating to a more recent version of Fedora Commons was discarded due to feature changes between Fedora Commons version 3 and 4, primarily the removal of support for OAI-PMH.[11] A new system had to be a drop-in replacement, independent of what solution was chosen. The following questions have been discussed about the repository system. It was not always easy to find a simple answer, often an initial goal was specified with future extensions in mind.

- *Who should be able to deposit data?* The main customers are the major internal projects German Reference Corpus (DeReKo) and Archive for Spoken German (AGD). They should be accommodated for from the start. Later it should also be possible for any researcher at the institute to deploy their data. Even external researchers offering relevant data should be able to store it with us.
- *Who should be able to access the data?* The majority of the data has to be protected either due to copyright or because it contains personal information, so only internal access is possible. The metadata is expected to be always publicly accessible. Among the data accepted at a later point there could be data that is either freely available or available to academic researchers.
- *What kind of data should be catered for?* In an initial phase the data provided by the two main projects DeReKo and AGD should be catered for. Later all relevant data should be considered. In the case of the IDS that means all data relevant to German studies as well as linguistics of German. The first two use few standardized file formats[12] while the latter could lead to a wide range of file formats, both for source material and annotation data.
- *Are there any quality requirements on the data?* Both DeReKo and AGD have their own requirements on data they accept. Consequently their data can be accepted without major quality checks. In the future policies and quality standards have to be established and enforced, especially for external data to be accepted.
- *What metadata is required?* Metadata has to be provided in the CMDI format (Windhouwer and Goosen 2022) to be integrated in the CLARIN infrastructure. The metadata provided by DeReKo and AGD can be converted automatically. For the future it has to be decided if there can be a strong requirement on CMDI metadata for all resources.
- *How are persistent identifiers handled?* Persistent identifiers can and should be handled outside the system but contained in the metadata. Handles[13] have been used previously but Datacite DOIs[14] could be preferred because they don't require hosting a resolver.

[9] https://www.coretrustseal.org/
[10] https://www.clarin.eu/content/clarin-centres
[11] This feature was added again at a later point but trust in the development process was damaged due to this decision
[12] E.g. DeReKo is encoded using a dialect of TEI P5 called I5 (https://www.ids-mannheim.de/digspra/kl/projekte/korpora/textmodell)
[13] https://handle.net/
[14] https://datacite.org/

- *Is this repository independent or part of some larger infrastructure?* The IDS is a CLARIN centre. In addition, it is involved in the Text+ project[15] to build a national research data infrastructure for language data in Germany. Both infrastructures have their own requirements on interfaces and data to be provided. These requirements can mostly be handled by OAI-PMH[16] end-points. In the future an integration into a federated content search (FCS)[17] should be possible.
- *How should files be stored on the hard disk or other storage?* Data should be stored in a way that it can be understood without the repository interface.
- *How often is the data accessed?* Due to copyright and other restrictions, most of the data cannot be freely available. Consequently, the majority of the data can only be accessed internally which makes the repository mostly take on the role of a cold storage. In the future, with more publicly available data or more fine-grained permissions, the focus should be shifted towards a hot storage with means of public access.

To summarize: for the start we planned a system that can serve primarily as a cold storage for the research data at the institute. Furthermore, it serves as reference and documented evidence for all available data towards the research infrastructures we are involved in. Direct access to the data as well as accepting data from other sources is planned for the future.

2.2. Setup at the IDS

The initial plan at the IDS was to develop a completely new system. This bespoke system would have been based on the Oxford Common File Layout (OCFL)[18] to store the data. Due to slow progress and lack of human resources the plan was changed to build a repository based on Invenio RDM[19].

When the decision was taken to work with Invenio both our requirements and the Invenio RDM software had to be modified. The requirement of having a transparent method to store the data on disk had to be relaxed. Instead we have to rely on Invenio to interpret the stored data in the context of Invenio records. At the same time we require that our Invenio repository can be integrated into the research infrastructures named before. Invenio provides the necessary OAI-PMH interface (CERN, University, and contributors 2023b) but lacks support for CMDI metadata. Consequently we had to develop a plugin to add CMDI as a metadata format to the OAI-PMH provider[20]. Furthermore, we created a previewer for CMDI metadata files[21] for the Invenio web interface.

Even though Invenio RDM provides a large set of features we decided to disable most of them for our initial use case. Primarily we disabled user management and login. As a consequence the front-end can only be used to access public data, i.e. all metadata as well as publicly available corpus data. Invenio records cannot be created or edited using the web front-end any longer.

[15] https://www.text-plus.org
[16] https://www.openarchives.org/pmh/
[17] https://www.clarin.eu/content/federated-content-search-clarin-fcs-technical-details
[18] https://ocfl.io/
[19] We use the terms Invenio and Invenio RDM interchangeably in this and the following sections
[20] https://github.com/daherb/invenio-oaiserver-cmdi
[21] https://github.com/daherb/invenio-previewer-cmdi

Instead we developed a separate process outside the Invenio system itself to ingest data which will be described in the following section. One major feature provided by Invenio we rely on is versioning of records and consequently of datasets.

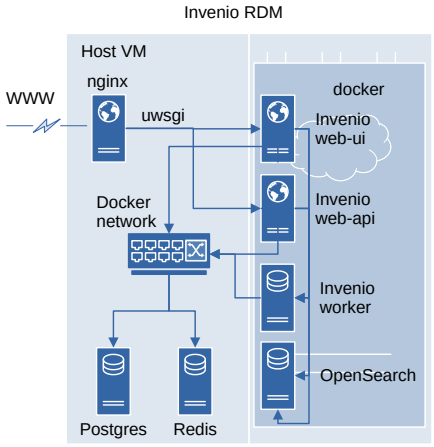


Figure 1: Deployment of Invenio RDM on the test system at the IDS. Invenio RDM and OpenSearch are dockerized and the database and web/proxy server are directly hosted on the host VM

We set up a multi-tier architecture with a development system, a test system and a production system. For the development system we followed the official installation instruction for a local install (CERN, University, and contributors 2022). For the test and production system we developed our own setup which can be seen in Figure 1.

2.3. Data Ingest Process

Besides installing and modifying the Invenio RDM system itself we focused our effort on a dedicated data ingest process, i.e. the way how research data is added to the repository. It is developed in Java as components of the corpus-services-ng[22] framework and interacts with Invenio using its REST API (CERN, University, and contributors 2023c). This API allows us, among others, to create and publish Invenio records, update record metadata, and add files to a record. Records are basic units within the Invenio repository and consist of metadata and a set of files. The metadata is aligned with DataCite's Metadata Schema v4.x (CERN, University, and contributors 2023a). Metadata is always public and all files of a record can either be public or private. For a more fine-grained permission management Invenio RDM would support "communities" which we will ignore here. Records are versioned and as soon as a record is published its files cannot be changed. The metadata can, however, still be updated.

The job of the ingest process is to take some suitable input and convert it into Invenio records. The process starts with a Submission Information Package (SIP). We settled on the BagIt (Kunze et al. 2018) format because it stores files together with a file list and checksums to verify file consistency. The bag contains the corpus data as well as metadata in CMDI format. Optionally, it can contain a file specifying how the files should be distributed across records

[22] https://github.com/daherb/corpus-services-ng

(*recordmap.xml*). Even though it is possible to store a complete dataset in a single record it makes sense to distribute one dataset across several records. This allows for example to update sub-corpora independently. An example SIP can be seen in Figure 3a.

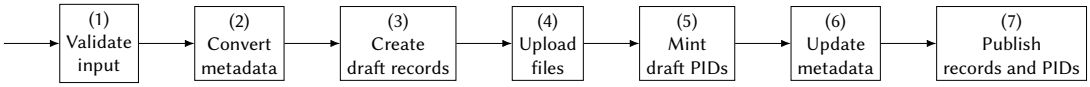


Figure 2: Simplified ingest pipeline showing all steps from the handling the input data to the published records and identifiers.

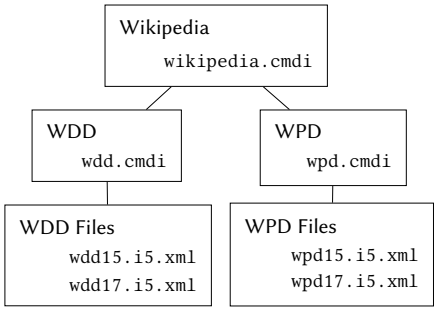
A simplified ingest pipeline is shown in Figure 2. The individual steps are:

(1) Before taking any further steps, basic validation happens on the input. At the moment only the consistency of the bag, i.e. the presence of all files and the correctness of the checksums is verified
(2) The metadata given as CMDI is converted into the Invenio-internal metadata format. Because the CMDI profiles are usually more expressive than the expected DataCite-compatible metadata the conversion does not cause any problems.
(3) A set of Invenio records is created in the draft state. Draft records can still be deleted if anything goes wrong. We decided to translate a dataset into a hierarchy of records (see Figure 3). This can be accomplished by creating links between records and allows us to separate metadata from data as well as private from public data. This can be helpful e.g. when updating data or when handling access.
(4) The files for the records are uploaded. After the upload the checksums can be validated once again.
(5) PIDs are created for the relevant records. We use DOIs and create them in the draft state to be able to delete them in case of a problem.
(6) The metadata, both the CMDI files and in the Invenio records, has to be updated to include the newly minted PIDs.
(7) Finally, if no errors occurred, the records and PIDs are published. From that point neither the records nor the PIDs that have been created can be deleted. To modify the files in a published record a new version has to be created. This even affects updating the CMDI metadata.

Using our Invenio test setup and ingest tool-chain we ingested three release versions of the DeReKo dataset. It resulted in three separate versions of the respective Invenio record. Data is deduplicated during the ingest process, i.e., files that are unchanged from the previous version are not uploaded again. Ingesting the first version (resulting in about 130 GB and 5,200 files) took a bit more than one hour. Ingesting the second version which was considerably larger (resulting in about 660 GB data and 19,200 files) took almost 6 h ours. The third version of similar size took about the same time (resulting in about 1.1 TB data and 31,500 files). A full performance analysis was not possible because the production system was not yet operational by the time of writing.

```
sip/
|-- data
|   |-- Metadata
|   |   |-- wikipedia.cmdi
|   |   |-- wdd.cmdi
|   |   \-- wpd.cmdi
|   \-- Content
|       |-- wdd15.i5.xml
|       |-- wdd17.i5.xml
|       |-- wpd15.i5.xml
|       \-- wdd17.i5.xml
|-- bagit.txt
|-- manifest-sha512.txt
|-- bag-info.txt
|-- tagmanifest-sha512.txt
\-- recordmap.xml
```

(a) Example SIP in the BagIt format



(b) Possible resulting record hierarchy

Figure 3: Example mapping from corpus files to Invenio records. The files of the dataset are split into several records that can be updated independently.

3. Discussion

This report is primarily focused on our own experience setting up a research repository. Part of this experience was the feeling of being alone with an enormous task and looking for other people who had a similar task and be able to compare their ideas with ours. With this report we hope we can help to fill this gap by sharing our experiences and decisions as well as the underlying reasons. We are aware that the system which we describe here is more of a starting point than the final result. Even a full description of what we did would unfortunately go beyond the scope of this article. But as long as we help any colleagues on their way we can consider it a success.

Our experience shows that it is a very challenging task to deploy your own research data repository. It costs a lot of time and/or financial resources and even requires in-depth knowledge and expertise. If the knowledge is not yet available it has to be acquired on the way, which slows down progress. This makes it seem like a task not worth its while for many institutions. This is especially true if other means of storing research data exist, being it as part of the local university or a larger research infrastructure. For example several university libraries provide access to research data repositories that can be used by all members of the university.

Despite this impression that it is unnecessary to host a local research data repository there can be both sufficient and necessary reasons to go this way. One sufficient reason could be the wish for data autonomy. As soon as you store your data with an external provider you have no direct control over permanent availability. One necessary reason to host data in a local repository is potential restrictions on the data itself. Depending on the data or its license it can only be stored and used in specific locations. This makes it impossible to transfer to external repositories. Similarly it is necessary to host a local repository if you require specific infrastructure and services that cannot be provided by an external repository. For example both CMDI metadata and the federated content search are very specific to the CLARIN ecosystem and as such only relevant to institutions working with language data.

If you wonder if it makes sense for you, the reader, to host your own research data repository, no final answer can be given. It really depends on the circumstances. However, if you have to walk this path we hope we can be of good company.

Acknowledgments

I would like to thank my colleagues in the Text+ project and especially Ines Pisetta who took over this challenging task when I moved on.

This work was funded by Text+, German Research Foundation (DFG) project number 460033370, as well as Nationella språkbanken, jointly funded by its 10 partner institutions and the Swedish Research Council (VR) (2018–2024; dnr 2017-00626).

References

CERN and OpenAIRE. 2013. *Zenodo.* https://doi.org/10.25495/7GXK-RD71.

CERN, Northwestern University, and contributors. 2022. *Install: Build, setup and run.* Accessed November 9, 2023. https://inveniordm.docs.cern.ch/install/build-setup-run/.

CERN, Northwestern University, and contributors. 2023a. *Metadata reference.* Accessed November 9, 2023. https://inveniordm.docs.cern.ch/reference/metadata/.

CERN, Northwestern University, and contributors. 2023b. *OAI-PMH.* Accessed November 9, 2023. https://inveniordm.docs.cern.ch/reference/oai_pmh/.

CERN, Northwestern University, and contributors. 2023c. *REST API reference.* Accessed November 9, 2023. https://inveniordm.docs.cern.ch/reference/rest_api_index/.

European Research Council Scientific Council. 2022. *Open Research Data and Data Management Plans.* Technical report. Version 4.1. https://erc.europa.eu/sites/default/files/document/file/ERC_info_document-Open_Research_Data_and_Data_Management_Plans.pdf.

Kunze, John A., Justin Littman, Liz Madden, John Scancella, and Chris Adams. 2018. *The BagIt File Packaging Format (V1.0).* Technical report RFC 8493. RFC Editor. https://doi.org/10.17487/RFC8493.

Swedish Research Council. 2022. *Vetenskapsrådets samordningsuppdrag om öppen tillgång till forskningsdata 2022.* Technical report. Stockholm, Sweden: Vetenskapsrådet (Swedish Research Council). https://www.vr.se/analys/rapporter/vara-rapporter/2022-03-10-vetenskapsradets-samordningsuppdrag-om-oppen-tillgang-till-forskningsdata-2022.html.

Windhouwer, Menzo, and Twan Goosen. 2022. "Component Metadata Infrastructure." In *The Infrastructure for Language Resources,* edited by Darja Fišer and Andreas Witt, 191–222. Berlin, Boston: De Gruyter. ISBN: 978-3-11-076737-7. https://doi.org/10.1515/9783110767377-008.

Żółtak, Mateusz, Martina Trognitz, and Matej Ďurčo. 2022. "ARCHE Suite: A Flexible Approach to Repository Metadata Management." In *Selected Papers from the CLARIN Annual Conference 2021,* edited by Monica Monachini and Maria Eskevich, 190–199. Linköping Electronic Conference 189. https://doi.org/10.3384/ecp18917.