NHR Data Management Systems Report

Marcel Nellesen (RWTH), Hendrik Nolte (GWDG), Christian Löschen (TUD)

July 2023

1 Introduction

Data management describes the process of storing, organizing and maintaining data. Effective handling of data is important to generate a long term value from the data. Data mangement systems (DMS) typically work on top of a data management platform. When considering research data the term research data management systems (RDMS) is used.

When focusing on research projects there are different motivations for research data management; ensuring good quality results, prevent knowledge loss from changing personal, allow reproducing results through external researchers, create a long term benefit of the granted funding.

Underlying these use cases are requirements that are often described as the FAIR data principles:

- **Findable:** To (re-)use data it must be possible to find it, therefore the data and metadata must be processable by humand and computers.
- **Accessable:** After finding the data, the user needs to know how to access it.
- **Interoperable:** The data is often integrated with other data, it must be possible to interoperate with tools and workflows for anaylsing, processing, and storing the data.
- **Reusable:** To reuse the data it is important that is described in detail, so it can be used in a different context.

This report contains a presentation of the different (R-)DMS solutions that are currently in use by the NHR centers. This is followed by a closer look on the lessons learned during the operation of these systems.

2 DMS Currently in Use

Within this section different DMS that are currently in use by the NHR centers are presented. For each of the solutions the following areas are adressed: a general overview of the platform, the provided functionality, the authentication and authorization infrastructure, available storage solutions, metadata handling, support for automation of processes and lastly the archiving of resources.

2.1 Coscine

At RWTH Aachen University, the research data platform Coscine has been developed as open source software since 2018 and is used for the management of research (meta)data, as well as for the allocation and provisioning of storage resources for research data. Coscine is developed according to the FAIR principles and implements interfaces for the so-called FAIR Digital Objects. For researchers, Coscine provides access to all research data of a research project, the linking with project- or subject-specific metadata as well as the management of project members. Thanks to its low-threshold access management, Coscine can be used as a collaboration platform beyond university boundaries. The development of Coscine also incorporates the findings and requirements from national (NFDI, NHR) and international projects (EOSC, gaia-x, RDA) and enables both subject-specific and cross-disciplinary use of the platform. As part of the Coscine.nrw service, the Coscine software is made available to all DH.NRW universities.

Coscine provides researchers with several different functionalities to manage their research data while following the FAIR-data principles. Coscine offers a project based structure for the management of project members, access rights, research data, and metadata. Through the integration of several different services the researchers are able to see and manage all of their research data in one place.

For authentification and authorisation Coscine uses the DFN-AAI. This is a central component of IAM4NFDI, thereby Coscine integrates into exisiting environement. For the authentification of external users, i.e. users that are not directly linked to a universityy or research facility within the DFN-AAI, it is possible to use the Open Researcher and Contributer ID (ORCID).

The access to storage space on the research data storage RDS.nrw is managed by a science-led application and review process. To ensure good scientific pactices (GSP) and the FAIR-principles the research data management concept must be described in research data management plan. In addition to RDS.nrw, GitLab (for text-based research data) or Linked Data codes (for metadata management and referencing of external data) resources can be created. Additionall (community-) cloud services, such as Sciebo and Nextcloud are planned.

The metadata management on project-, resource-, and file level is based on the W3C Standards RDF an the Shapes Constraint Language (SHACL) an ensures the interpretability and interoperability of the metadata. All elements that are managed by Coscine receive a persistant identifier (PID). Users can select from existing domain or project specific metadataprofiles or create their own. The creation of new profiles is possible through the inclusion of the AIMS project, even without deep knowledge in RDF and SHACL. Thereby, it is possible to reuse ontologies that were created by domain experts and consortia. For the interoperability with international infrastructures Coscine implements FAIR Data Point (FDP). Thsi is compatible with FAIR Digital Objects and is based od the Resource Definition Framework (RDF), the Data Catalog Vocabulary (DCAT) and the Linked Data Platform (LDP).

Coscine offers interfaces for the automation of workflows. With the REST-API, files can be automatically uploaded and linked with metadata. In addition it is possible to directly interact with the object storage with a S3 client, for large file transfers.

In additon to storing activly used, so called hot data, Coscine allows in combination with RDS.nrw the archiving of cold data an metadata for 10 years after the project ends. After this time period, a migration into a suitable long term archiving system is required.

Coscine is developed und a free license within a publically available gitlab project, which allows all interessted users to parttake in the development.

The available interfaces (REST APIs) and the continous integration with regongniced concepts like FAIR Digital Objects make Coscine an important part in developing distributed storage solutions according to the FAIR data principles. Through the metadata handling standarticed storage solutions can be used based on the FAIR data principles. This makes it easier for smaller universities to take part in the scientific infrastructures and led to a better scientific practice. The community driven development process allows many stakeholders to partake in the developement process. The software Coscine is currently being established in NRW as service Coscine.nrw, and is available to the universities in NRW. Additionally use cases outside of NRW from the NFDI consortia can use Coscine as a basis infrasturcture to manage their research data or research data storage.

2.2 iRODS

At JGU Mainz, iRODS was first introduced in 2019 as a long-term archiving solution for the entire university, with a particular emphasis on HPC-users. iRODS is intended as a solution to manage large scale data and its affiliated metadata. It has been developed according to the FAIR-principles allowing fine-grained access control ranging from local to worldwide access. Researchers benefit from the option to provide download links to direct (meta-)data access upon publication in journals.

iRODS provides, like Coscine, the possibility to manage a project based access control and (meta-)data handling. We at the JGU handle the access management through the internal Active-Directory (AD) Management. The creation of new HPC projects includes new projects within the AD, this includes external HPC user groups, too. Currently, iRODS is only provided to JGU-members (including the entire HPC user base, which in turn includes external HPC users). Nevertheless, to assist collaborations bidirectional access amongst partners can be granted using so-called iRODS-federations, if both institutions entertain iRODS-instances.

iRODS can manage a number of storage backends with varying numbers of replicas. Just recently, we migrated from Ceph with a replica on Tape (IBM's Spectrum Protect) to two copies at different locations on GPFS. The capacity is now 400 TB and easily extensible.

Internally, iRODS perceives metadata as so-called AVU-triplets: Attribute, Value, Unit. We provide a script to translate schema and XML files of arbitrary complexity to these AVU triplets, bidirectionally. A basic set of metadata (e.g. Author, Date, Publisher, Location and a default of ten years archiving time) is created automatically upon uploading datasets.

iRODS is command line based and can easily be integrated in ordinary HPC workflows. Additionally, using the already existing Python client, integration in data-centric workflow engines such as Snakemake and Nextflow is granted.

It is our intention to further enhance the usability of iRODS. For this we are in close collaboration with the University of Maastricht (NL). Current plans include better workflow integration, such as (semi-)automated DOI requesting. In a collaboration with the University of Trier iRODS should be integrated as backend for the web-based RDMS ViDa. Furthermore, there is a large community in the US and the Netherlands fostering the iRODS ecosystem.

Complementing the technical efforts, our institution strives to make the generation of data management plans (DMP) for HPC-users mandatory. To facilitate this process, we already maintain a RDMO-instance with DMP-templates tailored to different research subjects.

2.3 XNAT

Biomedical data produced by diagnostic techniques such as medical imaging and biosignal monitoring is extremely important in identifying and understanding many diseases. In the last decades, there has been steady progress in the clinical value of this data but also in required storage. Hence, to be able to develop new methods to analyze these large datasets, proper data management is key. During the last 20 years, XNAT, the *Extensible Neuroimaging Archiving Toolkit*, has established itself as a standard choice as a research data management system to organize physiological datasets, even beyond imaging techniques.

In XNAT, data can be organized in a hierarchical order. The highest level is a project, which is also used to manage user access. Within each project, subjects are organized, on which them experiments are being performed. An experiment can, for instance, be an imaging session, or the filling of a questionnaire. Each of these experiments can incorporate different data types again, e.g. in the case of an MRI scan, i.e., an image session, it can incorporate the raw scans, processed scans, and artifacts. In addition, using so called assessors, any kind of data can be associated with any kind of data type, including a project, experiment, subject, or scan. The already existing data types can be extended with custom data types. To do so, the attributes of the new data type has to be described in an *XML* document following the *XSD* format. Each data object, i.e., each concrete instance of a data type can be edited using the web interface of XNAT, e.g., to set the attributes.

Metadata is indexed in a *PostGreSQL* database. Each data type is represented as a unique table, where each attribute is a column. Therefor, using the web interface users can manually edit the indexed values. Through this interface, the data is also searchable, with respect to the access rights of the users, which are granted on a per project level.

User authentication and authorization can be managed by XNAT itself. For this the PostGreSQL database is used to manage the user information. At NHR@Göttingen we have integrated the XNAT user management into our SSO managed by GWDG, however restricted access in such a manne,r that each user has to be explicitly allowed by an admin. This gives the admins the chance, to clarify the general security level of the respetive XNAT instance and prevent sensitive data from being exposed in a purposely openly managed XNAT instance.

XNAT requires a POSIX-compliant storage interface. There are large-scale deployments, which use a fuse-based s3 mount point to store the data. At NHR@Göttingen we currently only use a *Cinder*-based and S3-backed POSIX-compliant Volume. Regular Backups are now written to an S3-Bucket, instead of an Tape-Archive. In the particular setup at NHR@Göttingen, the data directory is exported via NFS to a dedicated Docker-host, which us used for processing. The data can be accessed via the web interface, but also via a ReST API, bth times with respect to the users granted access.

XNAT has a rich history in including and providing processing and automation capabilities, which started with its own built-in workflow engine, dating back even beyond of the existence of XNAT itself. This was followed by the "Container Service", a mechanism to export data stored in XNAT into a Docker container, to execute arbitrary commands on the data. This service was continuously developed to support different backends, like Docker Swarm or Kubernetes. The workflow engine is known to be discontinued for years, so processing nowadays should be solely focused on the container service. The container service also offers a nice integration into the web UI of XNAT.

In order to serve even the most compute intensive tasks, a custom HPC service was developed at NHR@Göttingen in conjunction with the University Medical Center Göttingen. Here, the container service plugin is used to connect to HPCSerA, a ReST-based Function-as-a-Service interface developed at NHR@Göttingen. The container plugin only needs to send the required ReST-Call to HPCSerA, while it will take over the entire orchestration. An agent, which is periodically run on a frontend of Emmy, is pulling periodically the API server for new tasks. Once a new function call was made, it will be proxied to the agent, which then launches the function execution within the user space of each individual user. The execution is restricted to predefined functions, which can, however, be automatically be deployed into each users namespace. Data may be fetched by the function from the XNAT instance using the available ReST API. The HPCSerA agent then continuously monitors the state and can update it on both, the HPCSerA API server, and the XNAT instance. This allows for a secure job execution without any necessity for ssh access, on any of the involved components, reducing the attack surface drastically.

XNAT offers the possibility to archive projects. This, of course, also requires a proper setup to do so. At NHR@Göttingen it is currently investigated to move for archiving purposes to S3 from a Tape archive. It is anticipated, that recovery operations can be done much faster while S3 still offers a good cost to capacity ratio and a good data durability.

As XNAT promotes in its name, it is extensible. The two most prominent ways to extend the functionality of XNAT by the community is either the use of plugins, or by container commands. A plugin is a compiled, self-contained package, usually build with the Gradle build tool. These *jar* packages can then be shared by users in a market place, where numerous extensions have been posted. In addition, using container registries, such as Dockerhub, containe commands can be shared by users, allowing the easy portability for complicated workflows using things like *FreeSurfer*.

The current efforts at NHR@Göttingen focus on the inclusion of more functions into HPCSerA which are then in turn available for all XNAT users. This requires further investigation into more efficient data transfers and caching techniques with respect to the access control lists provided by XNAT.

2.4 DSpace

At TU Dresden, the research data repository service OPARA is based on DSpace. OPARA currently serves four Saxon universities. Its purpose is the archival and optional publication of research data.

DSpace is an open source software for repository services. It's main capabilities are customizable processes for the ingest of data and contextual information (metadata), accompanied by configurable multi-staged quality-assurance workflows. Therefore it is operated more than 3.000 times worldwide, predominantly by academic and research institutions, e.g. as publication server. Regarding the research data life cycle, DSpace is especially a service to store cold data at a stable, final state.

The content in DSpace is organized in communities, collections and items. Communities can contain other communities and are often organized hierarchically to reflect the organizational structure of an institution. Beside other communities, collections can also be part of every community. Collections may bundle a number of items and can not contain other collections or communities. An item represents a dataset, it may contain files and contextual information.

The usual user-centric submission process of new items is based on configurable web-forms. These can be composed to form sets of any complexity, complemented by a set of ready-to-use form sets like the Creative-Commons-License selector component. Beside the user-interactive submission, there are several options for automated data ingest, like the import of Archival Information Packages (AIP), ingesting data using the SWORD protocol or others.

Before new items are stored in DSpace, they do pass a submission workflow of automated and interactive tasks. Automated tasks can perform metadata-checks or -extraction, virus-scans or others. Interactive tasks are usually performed as part of possibly multi-staged quality assurance processes, where reviewers are either automatically assigned, can pick a task by themselves or perform a group-rating of an item.

Submission processes and -workflows can be defined collection-wise, which allows supporting different of these per DSpace instance, complemented by default configurations for collections without a dedicated submission process or workflow setup.

DSpace comes with an internal handle server, which assigns handles to every entity. Optionally, official handles can be assigned after registration of the service instance at handle.org. For items it is also possible to register DOIs. The included OAI-PMH server allows the automated harvesting of metadata of published information.

Users can authenticate by a number of different technologies, like simple local username+password login, by LDAP or Shibboleth identity provider, x509-certificates or by connecting to ORCID or OpenID as identity provider. Authorizations can be set up detailed on every entity (from communities up to single files) by roles and access policies for individual users and user groups.

Submitted files are internally stored either on local harddrives or an S3 storage. Metadata and internal data is stored on a relational PostgreSQL database. DSpace initially supports Dublin Core as basic metadata standard for bibliographic information and can be extended by additional metadata standards.

DSpace is developed as an open source technology. The project governance is organized under the hood of the LYRASIS non-profit organization. Its development relies on contributions of individual persons and institutions. The DSpace community and software development are organized by different boards, like the Steering Group, Leadership Group or the Community Advisory Team.

At TU Dresden the repository service OPARA is currently extended to better support ingesting and preserving large amounts of data. This is a current shortcoming of the DSpace software, but an essential requirement for the NHR center at TU Dresden. Also, more Saxon universities are currently on the loop to be served by OPARA and will be integrated soon.

3 Applicability for HPC and Future Challenges

Research data management is for many researchers a rather new topic. One of the curcial challenges that operators of RDMS are faced with is the adapability and acceptance of the platform. Researchers must have the possibility to easly include research data management into their daily work and make it an essential part of their workflows. This becomes especially important when HPC resources are involved. Within this sections a few challenges and possible solutions are discussed.

3.1 Research data management plan

3.2 Applying for storage space

To confront researchers early with research data management, it is possible to collect detailed information on the envisioned data and metadata handling strategies when they apply for storage space for their project. Here the RWTH Aachen uses the platform JARDS, that many researchers are already familier with from applications for computation time. A specialized Jards Instance was created that focuses only on storage applications.

Storage space is often required long before the first computations are performed on a cluster and to enforce a good scientific practise the data should be keept for at least 10 years, the time period for computation project is typically much shorter. Additionally, reviewing the storage application can be done by a specialized RDM team. They can support the researchers with the applications and the choice / creation of the right application schema, for their research project. If large amount of storage space are required a science let review process

3.3 Enforce good metadata handling

Many RDMS require researchers to define domain specific sets of metadata that describe their research data, while an extensive list of metadata fields can help to find and reuse research data, it requires the researchers to enter a large amount of metadata manually. This can be a barrier, as some researchers might not see the immediate benefit resulting in poorly described data.

Possible solutions would include the automated extraction of metadata from files as well as an extensive tool support for the researchers that makes entering the required informations easier.

3.4 Reuse by external researchers

RDMS should not only allow researchers to manage their own data, but also enable researchers from the same or different universities to discover interessting sets of research data that they would like do explore, access and reuse. This comes with it's own set of challenges, as externals would need easy access to the data and might, depending on the amount of data, require computation time on the local NHR center.

4 Conclusion

To ensure no important information about the research data is lost during creation, processing and analysing it is necessary to consider research data management from the start. Within this report we presented four different (research-) data mangement systems that are currently in use by the NHR centers. One of the big challenges is to integrate good researchdata management practices into the researchers workflow and let them see the benefit early on, for themselves and for future researchers, of using and maintaining their research data in a high quality. Existing process and workflows must be steamlined to ease the adaptation of these rdms and encourage the reusability.