GitLab as a tool for Research Data Management

Candidate for a Base4NFDI general service

Dirk von Suchodoletz[1][https://orcid.org/0000-0002-4382-5104], Dominik Brilhaus, [2][https://orcid.org/0000-0001-9021-3197], Marcel Tschöpe[1][https://orcid.org/0000-0002-3731-7664], and Jonathan Bauer[1][https://orcid.org/0000-0002-5624-2055]

[1] Computer Center University of Freiburg, Freiburg, Germany

[2] Cluster of Excellence on Plant Sciences (CEPLAS) Heinrich-Heine-University Düsseldorf, Germany

Abstract. The collaboration of researchers locally or worldwide ranging from single researchers over a group or a lab up to cross institutional and disciplinary cooperation requires suitable working environments. Data hubs in the form of science gateways--usually abstracting from just locally shared storage resources--are discussed and explored for quite some time. While it was en vogue to propose and develop discipline specific gateways, we suggest to rely on well-established standard software frameworks instead. Research data considered over the entire data life cycle and closely related activities like annotation, versioning and sharing has a lot in common with (open source) software development. Git and GitLab--well established tools in software development--could play a major role in general research data management. From DataPLANT's point of view, GitLab as a science gateway would be a valuable addition to the NFDI [5] software landscape. It would be beneficial to address it as a joint service in cross-domain activities of all interested consortia.

Keywords: Git, GitLab, versioning, sharing of data, science gateway, LFS, DataPLANT

Software development processes match central RDM requirements

Research data management (RDM) typically follows a cycle: data is gathered, analyzed, and published, leading to new research questions which often leads to a new start of the cycle or iterative loops. Anticipating the need for publication earlier in the process is beneficial, particularly in data gathering and analysis stages. For example, metadata annotation is crucial for data interpretation and reuse. Instead of annotating data just before publication, we propose continuous annotation throughout the RDM life cycle. Similarly, reproducibility should be continuous, ensuring that analyses on a data set are always reproducible.

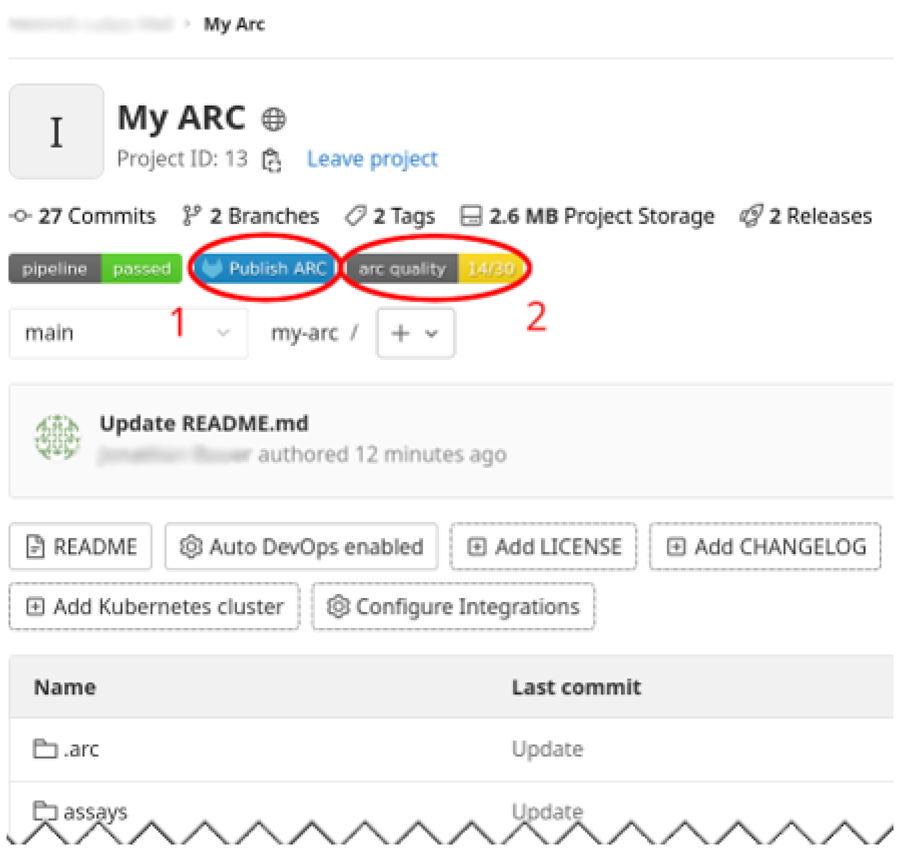
Rather than focusing on publication, research data can be managed and curated using processes similar to software development. This approach involves analyzing problems, designing solutions, and implementing and releasing iterations. Techniques like unit testing and continuous integration ensure the correctness of the data throughout the development phase and enable flexible release schedules. Many considerations in software development match the expectations in RDM quite well. The goal of each iteration of a cycle, i.e., producing a release, is anticipated throughout all steps. This has led to the development of techniques such as unit testing and continuous integration. [1]

Scientific research data is often seen as static and unchanging, but it should be viewed as dynamic and evolving. Version control can be a valuable tool for collaboration in RDM, allowing researchers to work together seamlessly while maintaining consistency, proper annotation, and reproducibility. By adopting version control as a systematic approach, ad hoc collaboration methods can be replaced, ensuring an atomic and unambiguous history of changes without burdening researchers. Automation, similar to software development practices, can be applied to RDM, enabling consistency checks, quality control, adherence to standards, and automated assessment of metadata and reproducibility. Version control also facilitates tracking the provenance of data and identifying contributors, streamlining the process of assigning credit and automating tedious tasks. The concept of sharing data throughout its evolution, as used in software development, applies to RDM, promoting collaboration and problem identification. Nevertheless, a couple of differences remain. While software development is primarily about highly structured text documents, we find all types of files in RDM. Admittedly, multiple adaptations are required to adapt software development toolchains for RDM.

GitLab and Git as the base of a data hub

The DataPLANT consortium [2] identified key requirements for a data hub in RDM, including versioning, group collaboration, multiple contributions, and easy access management. To fulfil these requirements, Git and the GitLab framework were chosen. Git provides versioning capabilities, allowing to track and undo / revert changes in a Git repository. DataPLANT offers an easy-to-use tool called ARC Commander[1] that abstracts low-level Git commands. Researchers can also use GitLab's interface or, the command line and other third-party Git integrations. Additionally, Git ensures file protection from accidental deletions and enables easy restoration of previous versions. The branching mechanism of Git is particularly valuable for the RDM lifecycle. It allows users to fork Annotated Research Context (ARC) [1] repositories, make modifications, and later merge them back into the original ARC repository. GitLab provides fine-grained access management, enabling users to form collaboration groups that they can manage themselves, fostering easy and flexible collaboration across institutions. To handle large data objects stored in ARCs, we utilize GitLab's built-in Large File Storage (LFS) server. LFS files are stored in an object store via S3 in the backend.

[1] https://github.com/nfdi4plants/arcCommander and the easy-to-use tool: ARCitect (i.e. GUI to ARC Commander) https://github.com/nfdi4plants/ARCitect

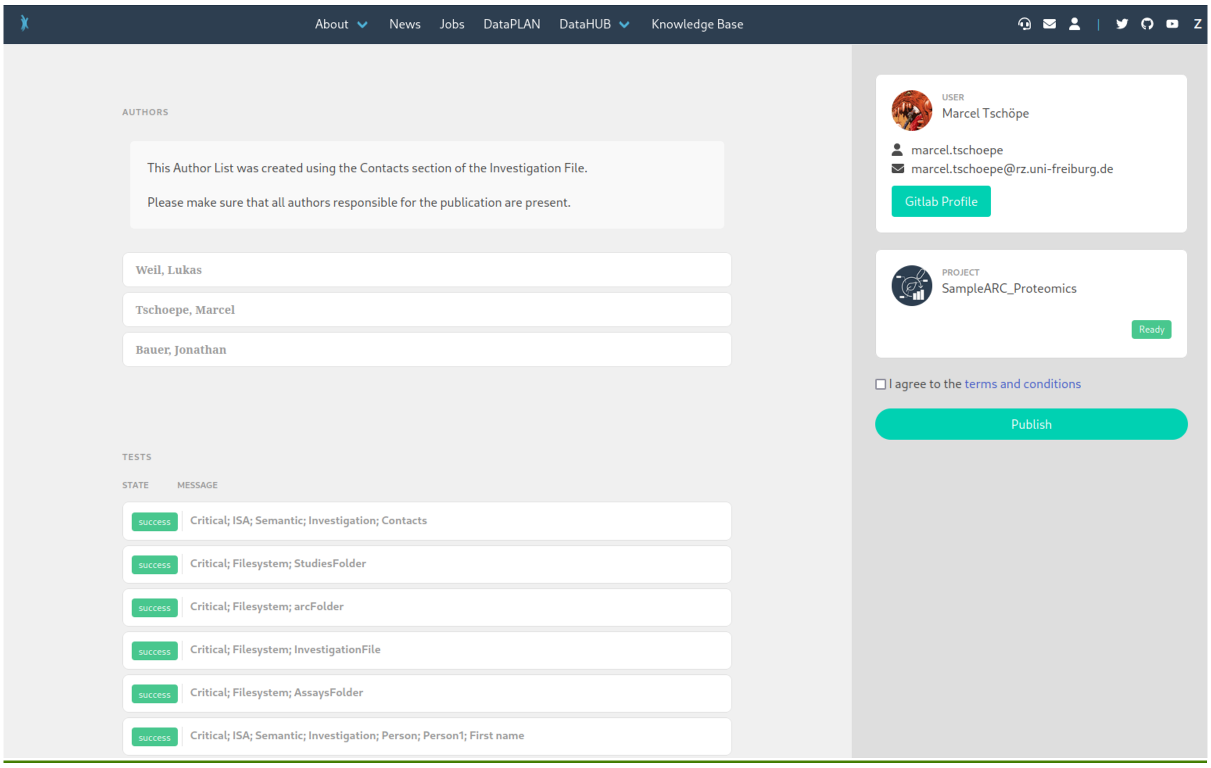


Name	Last commit
.arc	Update
assays	Update

Figure 1. The GitLab interface showing specific badges in the process of ARC management, quality assurance and preparation for data publication.

In DataPLANT, user management relies on established Authentication and Authorization Infrastructures (AAIs), such as Life Sciences AAI and ORCID, in combination with local authentication within the central DataPLANT authentication service. KeyCloak, developed by Red Hat and supporting modern authentication protocols like OpenID-Connect and SAML, is used by the underlying infrastructure for this purpose. It allows for the integration of multiple AAIs and handles the complex topic of identity brokering.

By implementing an AAI identity management system connected to GitLab and other services through these protocols, user management is simplified. KeyCloak enables the community to use their existing accounts, such as from their home institution, Life Sciences AAI, ORCID iD, or a future NFDI AAI. Different roles and permissions can be derived from the account source or specific attributes. Privileged users have full access to data and the ability to create archives/publications, while underprivileged users may have limited reporting capabilities or read-only access to raw data. It is important to note that these features are in the early stages and will be refined based on feedback from productive use of the infrastructure.



AUTHORS

This Author List was created using the Contacts section of the Investigation File.

Please make sure that all authors responsible for the publication are present.

Weil, Lukas

Tschoepe, Marcel

Bauer, Jonathan

TESTS

STATE	MESSAGE
success	Critical; ISA; Semantic; Investigation; Contacts
success	Critical; Filesystem; StudiesFolder
success	Critical; Filesystem; arcFolder
success	Critical; Filesystem; InvestigationFile
success	Critical; Filesystem; AssaysFolder
success	Critical; ISA; Semantic; Investigation; Person; Person1; First name

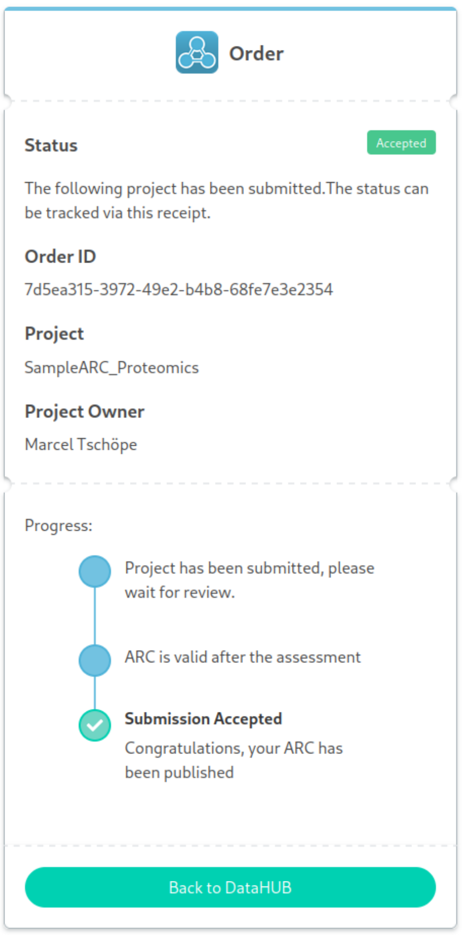
USER
Marcel Tschöpe
marcel.tschoepe
marcel.tschoepe@rz.uni-freiburg.de
Gitlab Profile

PROJECT
SampleARC_Proteomics
Ready

☐ I agree to the terms and conditions

Publish

Figure 2. After pressing the publish button, the user is shown an overview of the publication. If all tests are successful, the user can publish the ARC.



Status

The following project has been submitted.The status can be tracked via this receipt.

Order ID

7d5ea315-3972-49e2-b4b8-68fe7e3e2354

Project

SampleARC_Proteomics

Project Owner

Marcel Tschöpe

Progress:

Project has been submitted, please wait for review.

ARC is valid after the assessment

Submission Accepted

Congratulations, your ARC has been published

Figure 3. After submitting, the user gets an overview of the current status of their publication.

Adaptation of GitLab to RDM needs

GitLab is a versatile framework that offers numerous features and user interfaces, making it applicable beyond the software development community. In particular, agile project management features align well with the needs of daily (laboratory) RDM. Issue lists and boards help to organize project tasks. An associated, but technically independent wiki – with the same access permissions as the ARC – helps to communicate ideas, drafts, notes and meeting minutes, while keeping the ARC clear of indirect research data items. The GitLab's web IDE allows to flexibly contribute to the ARC or wiki on the fly.

As an open-source project, GitLab can be customized to meet the specific needs of RDM. However, certain advanced features, such as automatic merging of requests and code quality checks, are only available in the premium version of GitLab. Dealing with these limitations using custom tools and procedures can be time-consuming, and there may be uncertainty about the compatibility of code changes with the original open source GitLab version.

To address these challenges, some features can be extended with custom services. For example, a microservice can be developed for the ARC publication workflow due to the absence of external credentials support in the open source GitLab version. Additionally, the publication workflow developed within DataPLANT, utilizing GitLab's CI/CD pipelines and publication badge (figure.1), was not intuitive. To increase usability, a publishing microservice was integrated into GitLab as an OAuth application (Fig.2, Fig.3). This application would obtain an access token for the logged-in user and provide an overview of available ARCs for publication.

To avoid clogging the Git database with large binary files and render it unusable, we deploy mitigation strategies like Large File Support. Implementing GitLFS effectively in GitLab presented challenges due to certain built-in limitations which usually do not get hit in standard software development. Initially, researchers faced issues when attempting to upload files larger than 20 GB, prompting us to conduct a thorough investigation into the object limits within larger ARCs. Our findings confirmed that the production system indeed had a limit around 20 GB. Additionally, there were timeouts encountered when handling large files over wide-area network (WAN) connections, which can be addressed through patches.

However, there is a limitation related to the number of chunks and chunk sizes, preventing the upload of files larger than 50 GB. This limitation arises from a precompiled Go language component in GitLab responsible for multipart uploads into S3. To work around this constraint, the upload is first performed into a temporary object, and then a Ruby server-side copy process transfers it to its final storage destination. This process introduces similar but slightly different limitations.

Conclusion and Outlook

GitLab in DataPLANT has been in productive operation approximately for two years now, hosting 177 users and managing 241 ARCs, which collectively contain approximately 9.1 TB of data. As a powerful software development framework, it offers a wide range of features and user interfaces that extend its utility beyond the core software development community. As an open-source project, it can be customized to accommodate RDM requirements to a certain extent. At the same time, code changes from the open-source GitLab version require patching. Maintenance in general will benefit from the widespread nature of the selected components and an increasing pool of skilled potential personal filling respective positions.

However, certain features, such as automatic merging of requests with code quality checks and automatic merge approval rules, are only available in the premium version of GitLab. Although time-consuming, to address these limitations, some features can be replaced with custom services. For this reason, we have developed an external microservice for the ARC specifications, which allows users a secure publication.

Data availability statement

The material is available from public source: https://github.com/nfdi4plants/DataHUB and the Docker-Image: https://github.com/nfdi4plants/DataHUB/pkgs/container/datahub

Author contributions

-

Competing interests

-

Funding

Please insert a funding statement (if applicable) here.

Acknowledgement

We acknowledge the support of DataPLANT (NFDI 7/1 – 442077441) as part of the German National Research Data Infrastructure. We thank the Ministry of Science, Research and Art in Baden-Württemberg for their support of the BioDATEN Science Data Center which provided the necessary means for the InvenioRDM workflow integration. We further acknowledge CEPLAS, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within Germany´s Excellence Strategy – EXC-2048/1 – project ID 390686111.

References

1. C. Garth, J. Lukasczyk, T. Mühlhaus, B. Venn, J. Kr ̈uger, K. Glogowski, C. Martins Rodrigues, und D. von Suchodoletz, “Immutable yet evolving: ARCs for permanent sharing in the research data-time continuum,” in E-Science-Tage 2021: Share Your Research Data, V. Heuveline und N. Bisheh, Hrsgg. Heidelberg: heiBOOKS, 2022, S. 366–373. [Online]. Available: https://doi.org/10.11588/heibooks.979.c13751
2. C. Martins Rodrigues, D. von Suchodoletz, T. M ̈uhlhaus, J. Kr ̈uger, and B. Usadel, ,,DataPLANT – Ein NFDI-Konsortium der Pflanzen-Grundlagenforschung,” 2021. [Online]. Available: https://bausteine-fdm.de/article/view/8335
3. J. Bauer, M. Tschöpe, D. von Suchodoletz, C. M. Rodrigues, J. Weidhase, T. Mühlhaus, C. Garth, G. Doniparthi, H. Gauza, und L. Perelo, “From DataPLANT’s DataHUB to DataPUB(lication),” 2023, Accepted for publication by the 15th International Workshop on Science Gateways (IWSG2023), 13-15 June in Tübingen.
4. C. T. Jacobs and A. Avdis, ,,Git-RDM: A research data management plugin for the Git version control system,” Journal of Open Source Software, Vol. 1, Nr. 2, S. 29, 2016. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.00029.pdf
5. N. Hartl, E. Wössner, and Y. Sure-Vetter, ,,Nationale Forschungsdateninfrastruktur (NFDI),” Informatik Spektrum, Vol. 44, S. 370–373, 2021. [Online]. Available: https://doi.org/10.1007/s00287-021-01392-6
6. B. Venn, K. Schneider, K. Frey, H. L. Weil, J. Werner, F. Wannenmacher, T. Zajac, D. von Suchodoletz, B. Usadel, und J. Krüger, “Fostering the democratization of research data by using the Annotated Research Context (ARC) as practical implementation,” in E-Science Days 2021. Heidelberg: Universitatsbibliothek Heidelberg, 2021. [Online]. Available: https://doi.org/10.11588/heidok.00029769