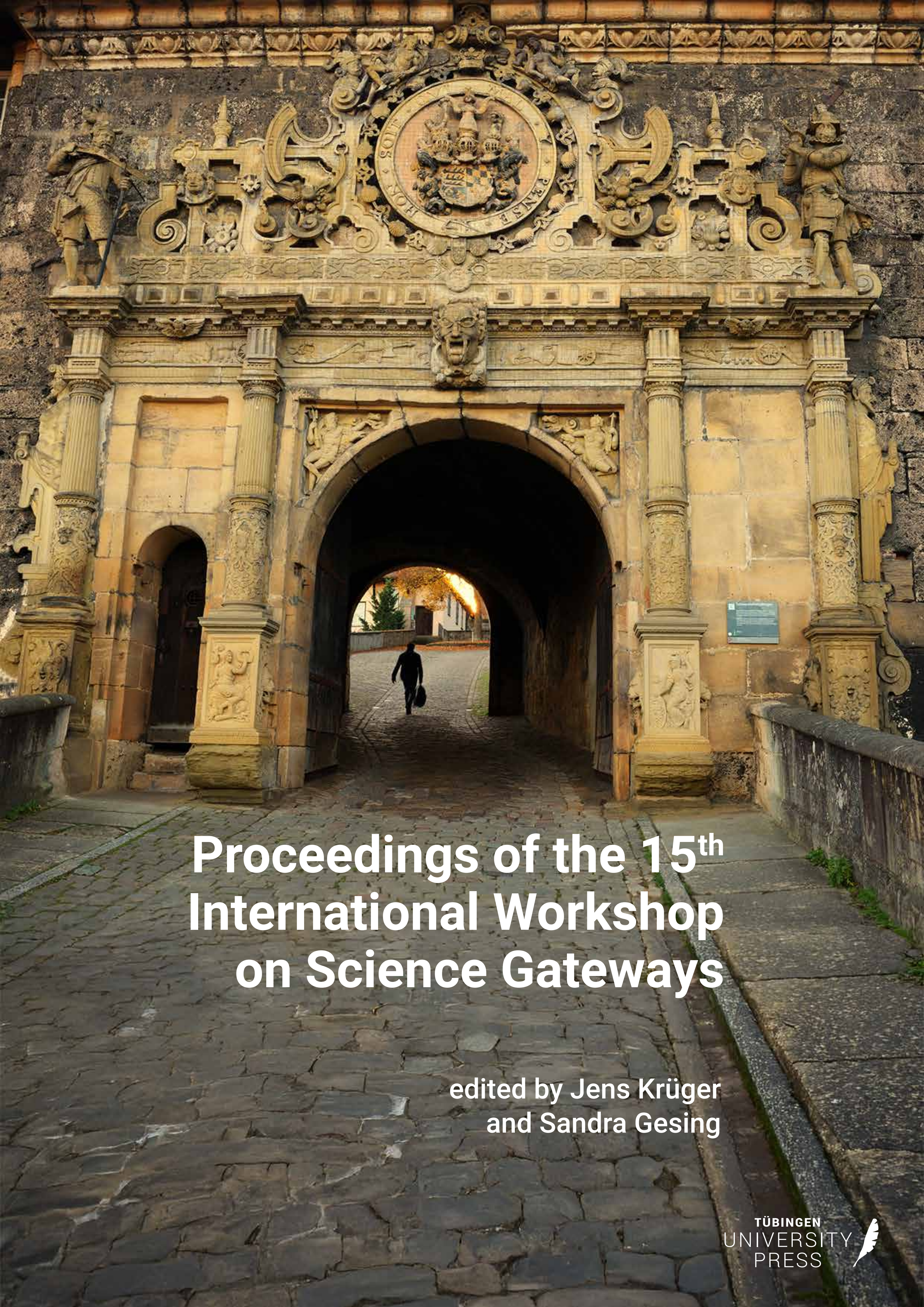


Proceedings of the 15th International Workshop on Science Gateways

edited by Jens Krüger and Sandra Gesing

TÜBINGEN
UNIVERSITY
PRESS

PROCEEDINGS
OF THE 15TH INTERNATIONAL WORKSHOP
ON SCIENCE GATEWAYS (IWSG2023)

Proceedings of the 15th International Workshop on Science Gateways (IWSG2023)

edited by
Jens Krüger and Sandra Gesing

TÜBINGEN
UNIVERSITY
PRESS

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie, detaillierte bibliografische Daten sind im Internet über http://dnb.de abrufbar.

Dieses Werk ist unter einer Creative Commons Lizenz vom Typ Namensnennung – Nicht kommerziell n 4.0 International zugänglich. Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie http://creativecommons.org/licenses/by-nc-nd/4.0/ oder wenden Sie sich brieflich an Creative Commons, Postfach 1866, Mountain View, California, 94042, USA.

Die Online-Version des vollständigen Manuskripts dieser Publikation ist auf dem Repositorium der Universität Tübingen verfügbar unter:
http://hdl.handle.net/10900/158990
http://nbn-resolving.de/urn:nbn:de:bsz:21-dspace-1589908
http://dx.doi.org/10.15496/publikation-100323

Tübingen University Press 2024
Universitätsbibliothek Tübingen
Wilhelmstr. 32
72074 Tübingen
tup@ub.uni-tuebingen.de
https://tuebingen-university-press.de

ISBN (PDF): 978-3-98945-019-6

Satz und Umschlagsgestaltung: Sandra Binder, Susanne Schmid, Universität Tübingen
Coverfoto: Jörg Weyhe, unterer Bildrand erweitert mit der generativen KI von Photoshop

TABLE OF CONTENTS

PREFACE

The 15th International Workshop on Science Gateways (IWSG 2023) was held at the Eberhard Karls Universität, Tübingen, from June 13 to 15, 2023. As the scientific community continues to push the boundaries of knowledge and innovation, the importance of effective, collaborative, and accessible research tools becomes ever more critical. The IWSG stands as a beacon in this landscape, highlighting the transformative power of science gateways in enhancing research productivity and collaboration across disciplines and geographies. Science gateways—virtual environments that provide researchers with integrated access to data, computational resources, and collaboration tools—are revolutionizing how science is conducted. They democratize access to high-performance computing, large-scale data analysis, and specialized software, making it possible for researchers from diverse fields and institutions to participate in cutting-edge science. The IWSG 2023 brought together researchers from various domains and communities dealing with science gateways, research data management, and related topics.

The IWSG 2023 featured sessions dedicated to the latest developments in gateway technologies, case studies showcasing successful implementations, and reports on building and enhancing these tools. By bringing together a diverse group of stakeholders, including developers, researchers, and users, the workshop fostered a community that is not only technologically adept but also inclusive and collaborative.

The workshop was highlighted by two inspiring keynotes given by Sonja Herres Pawlis about the evolution in the chemistry community »From MoSGrid via ERflow and MASi to NFDI4Chem and DALIA: a personal perspective« and by Gerhard Klimek about »Driving Sustainability through Expanding into an Adjacent Field and with a Customer Relationship Management System (CRM): from nanoHUB.org to chipshub,« highlighting the development in the semiconductor community. The scientific program consisted of six full-paper presentations complemented by nine lightning talks. The speakers presented insights from various projects and initiatives covering various communities and technologies.

The scientific program was complemented by a Stocherkahn tour on the beautiful river Neckar and concluded with a fabulous dinner at the Weinstube Forelle, giving all participants and friends of the IWSG a chance to exchange ideas and thoughts in a relaxed environment.

While the benefits of science gateways are clear, the path forward is not without challenges. Issues such as cybersecurity, data privacy, and the need for sustainable funding models are critical topics that are addressed at the IWSG workshops. Ensuring that science gateways remain secure, reliable, and accessible while also being adaptable to the rapidly evolving technological landscape is a delicate balance that requires ongoing attention and innovation.

The editors
Jens Krüger
Sandra Gesing

For further information please visit the workshop website at:
https://iwsgateways.github.io/iwsg2023

CHAIRS

Jens Krüger, *Eberhard Karls Universität Tübingen, Germany*
Sandra Gesing, *San Diego Supercomputer Center and US Research Software Engineer Association, USA*

LOCAL ORGANIZER

Suvasini Thangaraj, *Eberhard Karls Universität Tübingen, Germany*
Holger Gauza, *Eberhard Karls Universität Tübingen, Germany*
Ursula Eberhardt, *Eberhard Karls Universität Tübingen, Germany*

PROGRAMME COMMITTEE

Malcolm Atkinson, *The University of Edinburgh*
Antun Balaz, *Institute of Physics Belgrade*
Leonardo Candela, *ISTI—CNR*
Leyla Jael Castro, *ZB MED Information Centre for Life Sciences*
Neil Chue Hong, *Software Sustainability Institute, EPCC, University of Edinburgh*
Daniele D'Agostino, *Università di Genova*
Rafael Ferreira da Silva, *Oak Ridge National Laboratory*
Sandra Gesing, *University of Illinois Discovery Partners Institute*
Keith Jeffery, *Keith G Jeffery Consultants ordinary PC member*
Joohyun Kim, *Division of Genetic Medicine, Vanderbilt University Medical Center*
Tamas Kiss, *University of Westminster*
Jens Krüger, *University of Tübingen*
Robert Lovas, *MTA SZTAKI—LPDS*
David Meredith, *STFC*
Christian Page, *CERFACS*
Dana Petcu, *West University of Timisoara*
Susana Sanchez, *Instituto de Astrofisica de Andalucía*
Cevat Sener, *METU*
Luca Trani, *KNMI*
Chen Wang, *CSIRO*
Eric Yen, *Academia Sinica Grid Computing Center*

ABSTRACT

The conferences hosted by the *International Workshop on Science Gateways* (IWSG) have a long-standing tradition. The workshop series aims to advance in the field of science gateways and to improve and make services more accessible to researchers in various fields. The IWSG 2023 included six full-paper presentations and was complemented by nine lightning talks. These contributions spanned multiple fields, from biology to astronomy and beyond, showcasing a variety of tools and methodologies.

In this talk, the evolution from *MoSGrid to NFDI4Chem* is presented. Starting in the early 2000s, when chemistry grid computing through the *Molecular Simulation Grid* (MSG) advanced the field. These advancements laid the foundation for today's research data management (RDM) initiatives. Germany's *NFDI4Chem* builds on this legacy by creating a national RDM infrastructure that includes training programs, tools for data standards, and electronic lab notebooks. This notable talk reviews 15 years of progress in chemoinformatics and future developments.

The *Virtual Environment for Research Data and Analysis* (VERDA) presented a talk on the Science Gateway under development for a Collaborative Research Center in Biology. VERDA integrates FAIR data principles and cloud computing to support nearly 20 subprojects with advanced *omics and imaging analysis capabilities.

Another contribution explored the potential of *Function-as-a-Service* (FaaS), particularly for data distribution and processing in distributed environments such as the *SKA Regional Center Network* (SRCNet). By deploying key radio interferometry workflows on FaaS platforms, this model showed promise for handling massive-scale scientific data processing.

The talk by *German Human Genome-Phenome Archive* (GHGA), aims to facilitate the sharing of sensitive human *omics data using FAIR principles. GHGA contributes to the broader *Federated European Genome-Phenome Archive* (FEGA), enhancing data accessibility for international research while prioritizing data privacy.

Further innovations were presented in incentivizing data sharing from *Internet of Things* (IoT) devices for scientific research through a smart contract-based framework. This system rewards users for contributing wearable device data to science gateways using distributed ledger technology.

The Sustainability Program initiated by the United States *Science Gateways Community Institute* (SGCI) and the *Australian Research Data Commons* (ARDC) provides training and support for ensuring long-term project viability in academic settings. By incorporating business-like strategies, project teams were equipped with the best practices for gateway user interfaces and sustainability planning.

Lastly, the *DataPLANT* consortium provides a Science Gateway to enhance data management in fundamental plant research. Its *DataHUB* supports research throughout the data life cycle, offering workflows for tasks like data annotation, structuring, and publication. This enables more efficient and accessible research data handling, inspired by software engineering principles.

In conclusion, *IWSG 2023* showcased cutting-edge technologies and methodologies that transform how scientists access, manage, and share data. These tools advance research capabilities and emphasize the need for sustainability, interoperability, and collaboration across various scientific disciplines.

On June 15, 2023, Attendees of IWSG 2023 were pictured in front of Alte Aula, Tübingen.

FROM MOSGRID VIA ERFLOW AND MASI TO NFDI4CHEM AND DALIA: A HISTORICAL PERSPECTIVE

ALEXANDER HOFFMANN, JOCHEN ORTMEYER, FABIAN FINK, SONJA HERRES-PAWLIS*
Institute of Inorganic Chemistry, RWTH Aachen University, Landoltweg 1a, 52074 Aachen
*sonja.herres-pawlis@ac.rwth-aachen.de

ABSTRACT

In the beginning of the 21st century, grid computing gained large importance for science. Chemistry was pioneering here with the Molecular Simulation Grid (MoSGrid) which had a visionary concept but not enough resources. Many of the ideas from those days have been translated via several further projects in changing colours of chemical workflows into modern research data management (RDM). Here, the national research data management initiative NFDI4Chem is combining and further developing key RDM tools such as repositories, minimum standards and electronic lab journals to a unique national research infrastructure. Key to the success of RDM in chemistry is also the development of suited training concepts and materials. The talk will highlight the development of user-friendly chemoinformatics tools at the frontier between chemistry, informatics and data science in the last 15 years and give perspectives on the next decade.

Keywords: *Workflows; Research Data Management; FAIR data; Electronic Laboratory Notebook; Chemotion*

1. RESULTS

Coming from a bioinorganic background and also working in bioplastics, our group always relied on extensive synthetic lab work in combination with multi-method spectroscopic characterization and theoretical analyses. However, in the early years, performing theoretical calculations always had technical hurdles for many chemists. In a collaboration with A. Brinkmann (now JGU Mainz), we came across the Grid concept and learned about the advantages of workflows in chemical computing. This led to the Molecular Simulation Grid (MoSGrid) project which started in 2008[i],[ii].

The MoSGrid project aimed to develop a distributed computing infrastructure to support molecular simulations in the field of computational chemistry and materials science. The project sought to provide a user-friendly and flexible platform that could facilitate the execution of complex simulations on a variety of hardware architectures, ranging from small clusters to large-scale grids and clouds. The main objective of the MoSGrid project was to enable scientists to perform simulations on a large scale and to improve the reproducibility and reliability of their results, while reducing the time and effort required for setting up and running simulations. The project also aimed to promote the sharing and reuse of simulation data and software through the development of standardised formats and interfaces.

The aim of the community project MoSGrid (Molecular Simulation Grid) was to create competitive advantages through the grid for this industrial and scientific sector, which had not been covered by any community so far. The basis for this was provided by molecular and quantum mechanical calculations, which allow valuable insights into processes at the molecular and supramolecular level, which are often not accessible experimentally, but represent important bases for decision-making. However, such simulations are increasingly not feasible with local computing resources due to their complexity. The applications range from basic biomedical research to materials science and computer-aided drug design. The provision of an infrastructure of services for molecular simulations, annotation and their storage in the D-Grid thus represents an essential aspect for the sustainable development and promotion of the science and business location. The core task of MoSGrid was the development of grid services for the user group of molecular simulation tools. The distributed DGrid infrastructure should be used

http://dx.doi.org/10.15496/publikation-100323

to realise high-performance computing in the field of molecular simulations, the annotation of results with metadata and their provision for data mining and knowledge generation. Established programs from the field of computational chemistry, such as Gaussian, Turbomole, NWChem, Gromacs, Amber, CPMD, FlexX and others, serve as basic codes to be made available in the D-Grid.

An important component of the MoSGrid project was the development of chemical data repositories, which provide access to calculated molecular properties via a portal as well as »recipes«—standard methods for the most important applications—with the help of which jobs are generated and submitted to the grid (preprocessing and job submission). With the help of customised adapters, the recipes generate input files for various established applications from information on molecule, method, base set, etc. Input files for various established chemical simulation programs. The consortium developed a basic set of adapters for the most important programs and use cases. These are available to the entire community. Further adapters can be created, entered and extended by the users. The simulation results obtained are automatically extracted with the help of suitable parsers adapted to the special output formats of the various programs and checked for elementary plausibility (post-processing). If desired by the user, these results are transferred to collaborative data repositories of molecular properties. Suitable here are description languages (markup languages) and a »minimum information« concept. The minimum information concept is still a valuable concept with regard to standards in the 21st century[iii].

The data repositories were intended to provide easy access to information about jobs that have already been calculated and their results, additional knowledge gain through the creation of cross-references between different result data sets and the external referencing of simulation results.

We used the MoSGrid science gateway that allows users to conduct molecular simulation studies on a large scale right from the beginning. An example of such a study is the conformational analysis of guanidine zinc complexes, which serve as active catalysts in the ring-opening polymerization of lactide[iv]. Workflow technologies facilitate this large-scale quantum chemical study. In this example, 40 *conformers* were generated for two guanidine zinc complexes, and their structures were optimised using Gaussian03. The resulting energies were processed using the quantum chemistry portlet within the MoSGrid portal. All meta- and post-processing steps were also carried out within this portlet, which offers comprehensive workflow features implemented via WSPGRADE and submitted to UNICORE. The workflow is shown in Figure 1[v].

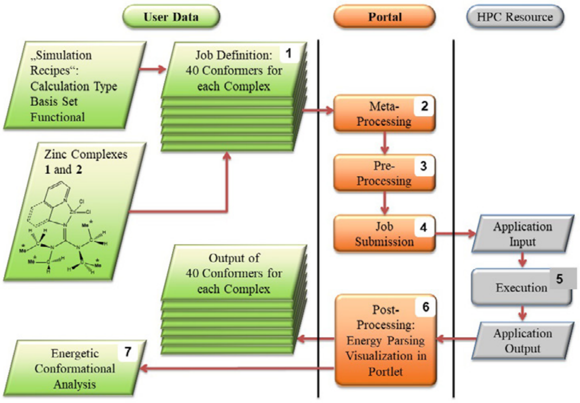

User Data
Portal
HPC Resource
„Simulation Recipes“: Calculation Type Basis Set Functional
Job Definition: 40 Conformers for each Complex
1
Zinc Complexes 1 and 2
Meta-Processing
2
Pre-Processing
3
Job Submission
4
Application Input
Execution
5
Output of 40 Conformers for each Complex
Post-Processing: Energy Parsing Visualization in Portlet
6
Application Output
Energetic Conformational Analysis
7

Figure 1: Quantum chemical workflow for zinc complexes[vi].

From 2012–2024 we participated to the ER-flow project which was building a European research community to advance workflow exchange and the study of scientific data interoperability in the workflow domain. The project built on the successes of the SHIWA project[vii]; in particular the coarse-grained workflow interoperability of the SHIWA simulation platform. It targeted key research areas that already use workflows for their regular experiments. The project consortium therefore included the scientific fields of astrophysics, computational chemistry, heliophysics and life sciences. To demonstrate the development, use and exchange of workflows, pilot workflows for specific use cases have been used, which were ported to the simulation platform within the framework of ER-flow and published in the workflow repository. On the one hand, the aim was to demonstrate how to use the simulation platform, and on the other hand, researchers can use the workflows for their experiments, whereby it has been possible to modify the existing workflows to create new ones. The pilot workflows will help to reach a critical mass of workflows to enable workflow exchange within and between communities. The project gathered and analysed community requirements for scientific data interoperability in the workflow domain. It examined existing protocols and standards. Within

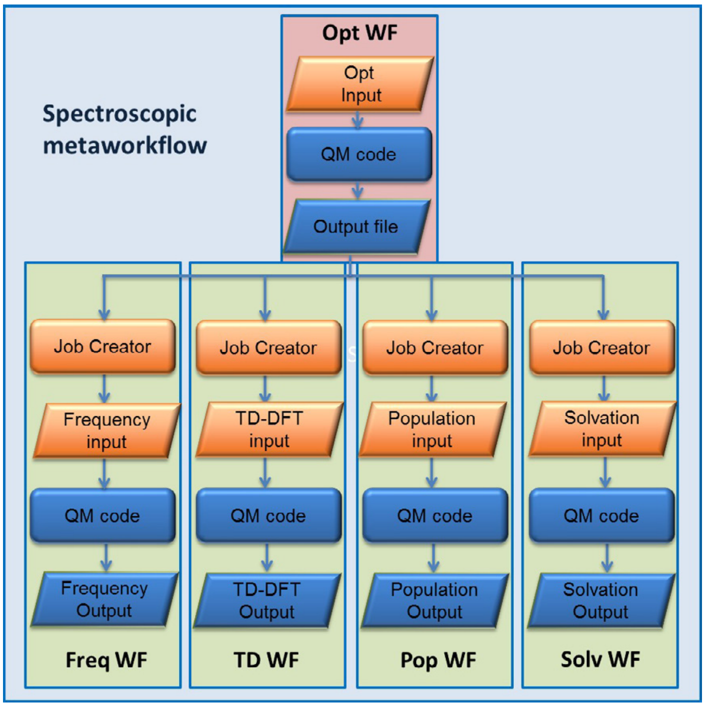


Figure 2: Spectroscopic metaworkflow[xxii],[xxiii].

ER-flow, we continued our computational studies on metal complexes[viii]. Methodologically, density functional theory is most appropriate here due to size of the system and investigated questions. The full simulation of molecular structures including the electronic structures comprises the calculation of optimised geometries, molecular orbitals, population analyses, frequencies, or optical absorptions. The combination of every one of these tasks as small basic workflow into a larger metaworkflow facilitates the simulators work enormously (Figure 2). This so-called spectroscopic workflow needs to be performed several times for an array of functionals and basis sets which have to be tested for the ultimate structural and optical description with regard to experimental data. Now, the spectroscopic metaworkflow can be combined into a new type of meta-metaworkflow with all being implemented in WS-PGRADE. This basic optimization (basic opt WF) serves as preoptimization step which saves calculation time in all subsequent optimizations included in the spectroscopic workflows (specX WFs). A meta-metaworkflow saves a lot of time in this application—more than a normal metaworkflow[ix].

Since 2015, we worked in the MASi (Metadata for Applied Sciences) project[x]. MASi is a modern repository service that was built based on the advanced KIT Data Manager (KIT DM) framework. The project aimed to provide a seamless and comprehensive data management system that can handle the data management requirements of various communities, both current and future. The goal was to offer a service that can manage living research data, which is becoming increasingly important in many areas of science and research. In the MASi project, we collaborated with cultural sciences, earth sciences and data sciences.

MASi aimed to provide excellent performance and scalability through an overall performance evaluation. The performance evaluation was an essential aspect of the project, as it demonstrates the ability of a complex yet seamless service, comprising multiple components, to provide effective management of research data. The evaluation included consideration and measurement of key performance aspects on a large scale to ensure that MASi can meet the data management requirements of a diverse range of communities[xi]. The MASi project sought to be a model for generic repositories, providing effective management of research data. The system was designed to handle the complexities of data management, including data discovery, access, sharing, reuse, and preservation. The goal was to provide a reliable, secure, and user-friendly service that can facilitate collaboration and accelerate research progress. One of the unique features of MASi was its advanced metadata management capabilities. The system could capture and store rich metadata about research data, which is critical for ensuring its discoverability and reuse. MASi was designed to integrate with existing research infrastructures and tools, enabling researchers to seamlessly manage their data across different platforms and systems. We were able to use MASi productively for a large kinetic study on entatic state copper complexes[xii].

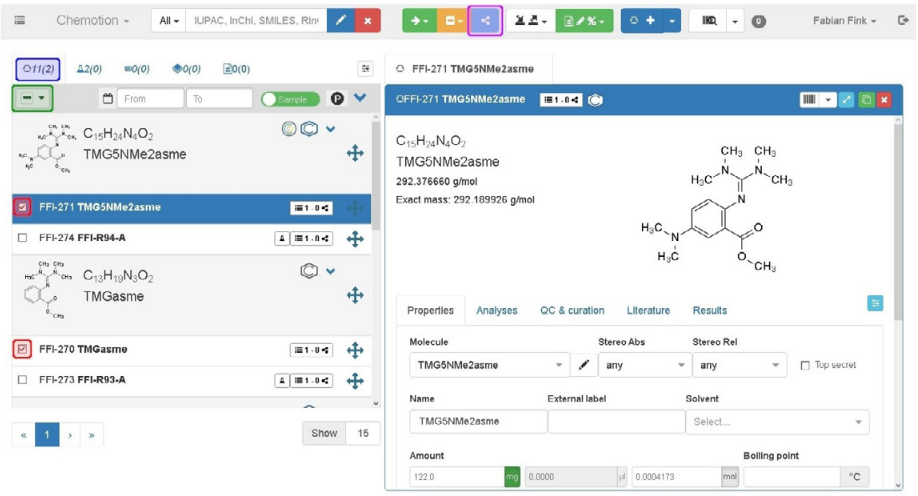
In summary, the MASi project represented a significant effort to develop a modern and comprehensive repository service for managing research data. By providing excellent performance and scalability, advanced metadata management capabilities, and seamless integration with existing research infrastructures, MASi contributed to transform the way research data is managed and shared.

In the end, MASi was only a puzzle piece on the way to something larger coming up in the German research landscape—the National Research Data Initiative (NFDI). Since 2020, we are active in the consortium dealing with molecular data, NFDI4Chem[xiii]. The goal of NFDI4Chem (National Research Data Infrastructure for Chemistry) is to establish a national research data infrastructure that serves the needs of the chemical sciences community in Germany. Specifically, NFDI4Chem aims to provide the infrastructure and services necessary for the storage, management, analysis, and sharing of research data in the field of chemistry, including data from experiments, simulations, and other types of research. The infrastructure provided by NFDI4Chem will support the entire research data lifecycle, from data acquisition and curation to preservation and reuse. This will help to improve the accessibility, interoperability, and quality of research data in chemistry, and facilitate collaboration and knowledge transfer across the field.

The chemical user community are an important part of the research data cycle in NFDI4Chem, as the users are the ones generating and working with the data. The infrastructure and services provided by NFDI4Chem are designed to support chemical users at all stages of the data cycle, from data generation and curation to analysis and sharing. At the data generation and curation stage, chemical users are able to contribute their research data to the NFDI4Chem infrastructure, where it can be stored securely and curated to ensure its quality and usability. NFDI4Chem provides tools and services for metadata creation, data cleaning, and data normalization, which help chemical users to make their data FAIR (Findable, Accessible, Interoperable, and Reusable)[xiv] and ready for sharing. At the analysis stage, chemical users are able to utilise the tools and services provided by NFDI4Chem for data processing, visualization, and modelling. Finally, at the sharing stage, chemical users are able to make their data available to the wider research community, either through public repositories or through controlled-access portals. NFDI4Chem provides services for data publication and citation, as well as tools for collaboration and data sharing within and across research teams. Overall, the integration of chemical users into the data cycle in NFDI4Chem ensures that the infrastructure and services provided are responsive to the needs of the research community, and that the data generated is of the highest quality and usability.

For molecular chemists, the electronic lab notebook (ELN) Chemotion[xv],[xvi] plays a key role to start the digital life cycle already during the experiment planning. We have integrated Chemotion ELN into the group's lab work already three years ago and also developed a working group policy[xvii]. Chemotion ELN is an open-source electronic laboratory notebook specifically designed for the chemical sciences. It offers several advantages for chemical users, including:

- Localised data storage: Chemotion ELN provides a local location for storing and organising research data. It also allows for version control, ensuring that previous versions of data are not lost or overwritten. Via the Chemotion Repository, the data can be deposited and exchanged with others.
- Collaboration: Chemotion ELN facilitates collaboration among research teams, as users can share data, annotations, and comments in real-time. This makes it easier for researchers to work together and share information, regardless of location or time zone.
- Data security: Chemotion ELN offers robust security features, such as encrypted data storage, access controls, and audit trails, to ensure that data is protected from unauthorised access or tampering.
- Customisable: Chemotion ELN is an open-source platform, which means that users can



$C_{15}H_{24}N_4O_2$
TMG5NMe2asme

☑ FFI-271 TMG5NMe2asme 1 - 0
□ FFI-274 FFI-R94-A 1 - 0

$C_{13}H_{19}N_3O_2$
TMGasme

☑ FFI-270 TMGasme 1 - 0
□ FFI-273 FFI-R93-A 1 - 0

FFI-271 TMG5NMe2asme

$C_{15}H_{24}N_4O_2$
TMG5NMe2asme
292.376660 g/mol
Exact mass: 292.189926 g/mol

Molecule: TMG5NMe2asme
Stereo Abs: any
Stereo Rel: any
□ Top secret
Name: TMG5NMe2asme
External label:
Solvent: Select...
Amount: 122.0 mg 0.0000 μl 0.0004173 mol
Boiling point: °C

Figure 3: User interface of the Chemotion ELN[xxiv],[xxv].

customise it to fit their specific research needs. It offers a variety of plug-ins and add-ons, such as chemical structure drawing tools, which can be easily integrated into the platform.

- FAIR compliance: Chemotion ELN is designed to comply with the FAIR principles (Findable, Accessible, Interoperable, and Reusable) for research data. This means that data stored in the Chemotion ELN and also in the Chemotion Repository is well-organised, documented, and easily accessible, making it easier to share and reuse.

Overall, the Chemotion ELN and Chemotion Repository offers chemical users an efficient and secure way to manage and share their research data, while also promoting collaboration and compliance with FAIR principles. The user interface (Figure 3 as example) guides the user through the synthesis and allows also the integration of analytical data[xviii,xix]. On the long range, a working group builds up its own database.

The training in Chemotion ELN provided by NFDI4Chem is driven by the goal of helping chemical users to effectively and efficiently use the platform to manage their research data. The training is designed to provide users with the knowledge and skills they need to fully utilise the features and functionalities of Chemotion ELN. The specific objectives of the training in Chemotion ELN include:

- Familiarising users with the platform: The training may cover the basic features and functionalities of Chemotion ELN, such as creating and managing experiments, adding data and annotations, and sharing data with collaborators.
- Providing guidance on best practices: The training may provide guidance on best practices for using Chemotion ELN, such as how to properly document experiments, organise data, and ensure data security and privacy.
- Addressing specific use cases: The training may be tailored to specific use cases, such as chemical synthesis or analytical chemistry, to ensure that users are equipped with the knowledge and skills they need for their specific research tasks.
- Integrating with other tools and platforms: The training can cover how to integrate Chemotion ELN with other tools and platforms commonly used in chemical research, such as chemical structure drawing tools or data analysis software.

Herefore, we have developed a set of training videos[xx], together with a Chemotion workshop concept and integrated this also in curricular training of chemistry students in Aachen. Besides, we also developed general research data management (RDM) training material for chemists to foster the cultural change towards a better digitization of chemistry.

Since most research data management materials available are rather generic, we got engaged in the NFDI section EduTrain[xxi] which sets out to collect, organise and provide learning materials to the community on all levels and for all disciplines (Figure 4).

Very recently, the Data Literacy Alliance project DALIA started which will form the basis for the practical work of the section.

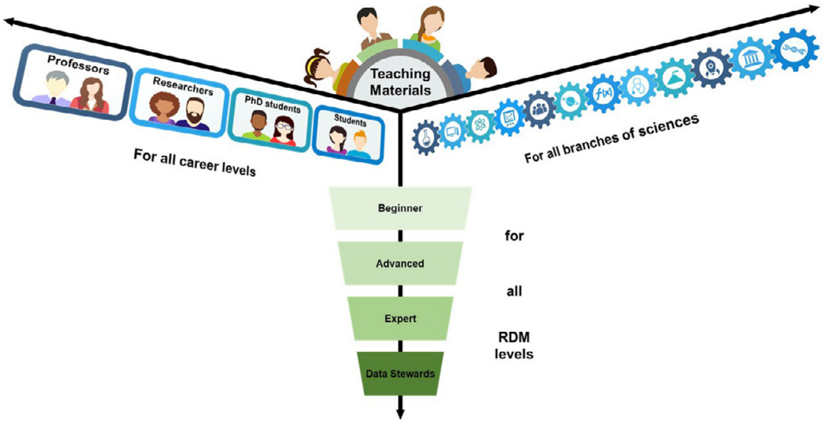


Figure 4: Dimensions of training material on research data management[xxvi]

2. CONCLUSION

In this talk, the evolution from MoSGrid to NFDI4Chem is presented and directions for future progress will be highlighted from the perspective of a user and early adopter. user-friendliness is key to all software tools since time and money are short in synthesis laboratories. Moreover, the cultural change needs more time in the research community. NFDI4Chem works on all aspects to lower technical hurdles by providing user-friendly tools such as ELNs and repositories but also mental hurdles via teaching on all levels. A further focus lies in the future development of the InChI towards inorganic chemistry to foster the utilization of digital molecular representations in inorganic chemistry. On the long term, this will enable machine learning in inorganic chemistry as well.

3. ACKNOWLEDGMENT

The authors thank the BMBF (Federal Ministry of Education and Research: MoSGrid project (FKZ: 01IG09006) and Knowledge-Graph der Data Literacy Alliance (Dalia) für FAIRe Datennutzung und -bereitstellung auf der Basis von Semantic-Web-Technologie project (FKZ:16DWWQP07B)), the Deutsche Forschungsgemeinschaft (MASi, project number 262962976 and NFDI4Chem, project number 441958208) and the European Commission (ER-Flow, FP7-Infrastructure; Grant agreement ID: 312579) for funding.

4. REFERENCES

i J. Krüger, R. Grunzke, S. Gesing, S. Breuers, A. Brinkmann, L. de la Garza, O. Kohlbacher, M. Kruse, W. E. Nagel, L. Packschies, R. Müller-Pfefferkorn, P. Schäfer, C. Schärfe, T. Steinke, T. Schlemmer, K. D. Warzecha, A. Zink and S. Herres-Pawlis, »The MoSGrid Science Gateway—A Complete Solution for Molecular Simulations,« *J. Chem. Theor. Comput.*, vol. 10, pp. 2232–2245, 2014. https://doi.org/10.1021/ct500159h

ii S. Gesing, R. Grunzke, J. Krüger, G. Birkenheuer, M. Wewior, P. Schäfer, B. Schuller, J. Schuster, S. Herres-Pawlis, S. Breuers, Á. Balaskó, M. Kozlovszky, A. S. Fabri, L. Packschies, P. Kacsuk, D. Blunk, T. Steinke, A. Brinkmann, G. Fels, R. Müller-Pfefferkorn, R. Jäkel, and O. Kohlbacher, »A Single Sign-On Infrastructure for Science Gateways on a Use Case for Structural Bioinformatics,« *J. Grid Computing*, vol. 10, pp. 769–790, 2012. https://doi.org/10.1007/s10723-012-9247-y

iii S. Herres-Pawlis, F. Bach, I. J. Bruno, S. J. Chalk, N. Jung, J. C. Liermann, L. R. McEwen, S. Neumann, C. Christoph, C. Steinbeck, M. Razum, and O. Koepler: Minimum Information Standards in Chemistry, »A Call for Better Research Data Management Practices,« Angew. Chem. Int. Ed., vol. 61(51), Art. no. e202203038, 2022. https://doi.org/10.1002/anie.202203038

iv I. dos Santos Vieira and S. Herres-Pawlis, »Lactide Polymerisation with Complexes of

Neutral N Donors—New Strategies for Robust Catalysts,« *Eur. J. Inorg. Chem.*, vol. 2012, pp. 765–774, 2012. https://doi.org/10.1002/ejic.201101131

v S. Herres-Pawlis, G. Birkenheuer, A. Brinkmann, S. Gesing, R. Grunzke, R. Jäkel, O. Kohlbacher, J. Krüger, and I. dos Santos Vieira, »Workflow-enhanced Conformational Analysis of Guanidine Zinc Complexes via a Science Gateway,« *Stud. Health Technol. Inform.*, vol. 175, pp. 142–151, 2012. https://doi.org/10.3233/978-1-61499-054-3-142

vi S. Herres-Pawlis et al., *Stud. Health Technol. Inform.*, vol. 175, pp. 142– 151, 2012.

vii SHIWA Project. »SHIWA: SHaring Interoperable Workflows for large-scale scientific simulations on Available DCIs.« [Online]. Available: http://observatory.rich2020.eu/rich/projects/view/261585.

viii A. Hoffmann and S. Herres-Pawlis, »Hiking on the Potential Energy Surface of a Functional Tyrosinase Model—Implications of Singlet, Broken-symmetry and Triplet Description,« *Chem. Commun.*, vol. 50, pp. 403–405, 2014. https://doi.org/10.1039/C3CC46893C

ix S. Herres-Pawlis, A. Hoffmann, T. Rösener, J. Krüger, R. Grunzke, and S. Gesing, »Multilayer Meta-metaworkflows for the Evaluation of Solvent and Dispersion Effects in Transition Metal Systems Using the MoSGrid Science Gateways,« *IEEE Xplore—2015 7th International Workshop on Science Gateways (IWSG)*, 2015, pp. 47–52. https://doi.org/10.1109/IWSG.2015.13

x R. Grunzke, V. Hartmann, T. Jejkal, H. Kollai, A. Prabhune, H. Herold, A. Deicke, C. Dressler, J. Dolhoff, J. Stanek, A. Hoffmann, R. Müller-Pfefferkorn, T. Schrade, G. Meinel, S. Herres-Pawlis, and W. E. Nagel, »The MASi Repository Service—Comprehensive, Metadata-driven and Multi-community Research Data Management,« *Future Gener. Comput. Syst.*, vol. 94, pp. 879–894, 2019. https://doi.org/10.1016/j.future.2017.12.023

xi R. Grunzke et al., *Future Gener. Comput.* Syst., vol. 94, pp. 879–894, 2019.

xii J. Stanek, N. Sackers, F. Fink, M. Paul, L. Peters, R. Grunzke, A. Hoffmann, and S. Herres-Pawlis, »Copper Guanidinoquinoline Complexes as Entatic State Models of Electron-Transfer Proteins,« *Chemistry A European J.*, vol. 23, pp. 15738–15745, 2017. https://doi.org/10.1002/chem.201703261

xiii C. Steinbeck, O. Koepler, F. Bach, S. Herres-Pawlis, N. Jung, J. C. Liermann, S. Neumann, M. Razum, C. Baldauf, F. Biedermann, T. W. Bocklitz, F. Boehm, F. Broda, P. Czodrowski, T. Engel, M. G. Hicks, S. M. Kast, C. Kettner, W. Koch, G. Lanza, A. Link, R. A. Mata, W. E. Nagel, A. Porzel, N. Schlörer, T. Schulze, H.-G. Weinig, W. Wenzel, L. A. Wessjohann, and S. Wulle, »NFDI4Chem—Towards a National Research Data Infrastructure for Chemistry in Germany,« *RIO*, vol. 6, Art. no. e55852, 2020. https://doi.org/10.3897/rio.6.e55852

xiv M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J-W. Boiten, L. Bonino da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Corsas, I. Dillo, O. Dumon, S. Edmonds, C. T. Evolo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Growth, C. Goble, J.S. Grethe, … and B. Mons, »The FAIR Guiding Principles for scientific data management and stewardship,« *Sci. Data*, vol. 3, Art. no. 160018, 2016. https://doi.org/10.1038/sdata.2016.18

xv P. Tremouilhac, A. Nguyen, Y-C. Huang, S. Kotov, D.S. Lütjohann, F. Hübsch, N. Jung, and S. Bräse, »Chemotion ELN: an Open Source Electronic Lab Notebook for Chemists in Academia,« *J. Cheminform.*, vol. 9, Art. no. 54, 2017. https://doi.org/10.1186/s13321-017-0240-0

xvi Electronic Laboratory Notebook (ELN) & Repository for Research Data. »Chemotion.« [Online]. Available: https://chemotion.net/

xvii F. Fink, H. M. Hüppe, N. Jung, A. Hoffmann, and S. Herres-Pawlis, Sharing is Caring: Guidelines for Sharing in the Electronic Laboratory Notebook (ELN) Chemotion as Applied by a Synthesis-Oriented Working Group, *Chem. Methods*, vol. 2, Art. no. e202200026, 2022. https://doi.org/10.1002/cmtd.202200026

xviii P. Tremouilhac et al., *J. Cheminform.*, vol. 9, Art. no. 54, 2017.

xix https://chemotion.net/

[xx] F. Fink, S. Benjamaa, N. Parks, A. Hoffmann, S. Herres-Pawlis, »Chemotion ELN Instruction Videos,« (Version 1.0, Feb. 13, 2023), doi: https://doi.org/10.5281/zenodo.7634481

[xxi] S. Herris-Pawlis, P. Pelz, N. Kockmann, R. Gläser, M. Richter, J. Liermann, J. Ortmeyer, I. Heine, A. Metzmacher, A-C. Andres, A. Münzmay, J-O. Heuer, M. Hagener, J. Dierkes, C. Wiljes, and B. Lindstädt, »Sektionskonzept Training & Education zur Einrichtung einer Sektion im Verein Nationale Forschungsdateninfrastruktur (NFDI) e.V.,« (Version 1.1, Oct. 26, 2021), doi: https://doi.org/10.5281/zenodo.5599770

[xxii] S. Herres-Pawlis, A. Hoffmann, A. Balasko, P. Kacsuk, G. Birkenheuer, A. Brinkmann, L. de la Garza, J. Krüger, S. Gesing, R. Grunzke, G. Terstyansky, and N. Weingarten, »Quantum chemical metaworkflows in MoSGrid,« *Concurrency Computat.: Pract. Exper.*, vol. 27, pp. 344–357, 2015. https://doi.org/10.1002/cpe.3292

[xxiii] S. Herres-Pawlis, A. Hoffmann, L. de la Garza, J. Krüger, R. Grunzke, S. Gesing, N. Weingarten, and G. Terstyansky, »Meta-metaworkflows for Combining Quantum Chemistry and Molecular Dynamics in the MoSGrid Science Gateway,« IWSG 2014 *6th International Workshop on Science Gateways*, Dublin, Ireland, 2014, pp. 73–78. https://doi.org/10.1109/IWSG.2014.20

[xxiv] F. Fink et al., *Chem. Methods*, vol. 2, Art. no. e202200026, 2022.

[xxv] F. Fink et al.,
doi: https://doi.org/10.5281/zenodo.7634481

[xxvi] S. Herris-Pawlis et al., doi: https://doi.org/10.5281/zenodo.5599770

COLLABORATION ACROSS BOUNDARIES: A SCIENCE GATEWAY WITH FEDERATED BACKEND

HALIMA SAKER, HOLGER GAUZA, JENS KRÜGER*, AMIR BALEGHI, ALEXANDER KIRBIS, SIMON PIRKL
High Performance and Cloud Computing Group, Eberhard Karls Universität Tübingen, Tübingen, Germany
*jens.krueger@uni-tuebingen.de

STEPHAN HACHINGER, ALEXANDER WELLMANN, JOHANNES MUNKE, MUKUND BIRADAR
Leibniz Supercomputing Centre (LRZ) Bavarian Academy of Sciences and Humanities Garching b.M., Germany

KLAUS F. X. MAYER
Plant Genome and Systems Biology Helmholtz Zentrum München, German Research Center
for Environmental Health Neuherberg, Germany

ABSTRACT

Science Gateways are designed to simplify the use of complex computational and data resources, allowing researchers to access powerful scientific computing resources, tools, and data without requiring advanced technical skills. In this contribution, we sketch an application of these concepts: building up a research data analysis and management infrastructure for a Collaborative Research Center in biology. The »Virtual Environment for Research Data and Analysis (VERDA)« we are constructing can be seen as a Science Gateway with a very rich backend. Conceptualized and implemented for the DFG-funded Collaborative Research Center »Genetic diversity shaping biotic interactions of plants (PlantMicrobe)«, it will serve almost twenty subprojects with data analysis in *omics, imaging, and beyond. Besides a cloud-computing environment, FAIR (Findable, Accessible, Interoperable, and Reusable) research data management infrastructure is a core feature of our program, as is dedicated data stewardship to warrant sustainability and user uptake. Collaborating with the National Research Data Infrastructure (NFDI) ecosystem, in particular with the DataPLANT consortium, VERDA will implement NFDI and ELIXIR methods and standards in a customized environment for a specific, very demanding target community.

Keywords: *Science Gateways, Matrix, GitLab, FAIR Data, Research Data Management, Federation, Cloud Infrastructure, Keycloak*

1. INTRODUCTION AND CONTEXT

Science Gateways and Virtual Research Environments (VREs) are well-accepted system concepts for supporting scientists in doing their work. The concrete systems are typically hosted on remote servers and provide researchers with access to scientific data in specialized repositories plus powerful, advanced computing resources such as high performance computing clusters and Infrastructure-as-a-Service (IaaS) clouds. Approaches to establishing such environments have often failed in the sustainability and continued-usage phase[i]. One key to success is certainly a design not too much oriented at IT principles and idealized research processes, but aimed at enabling scientists to adapt the environment to their research. Modern projects may thus implement a flexible research environment in collaboration with researchers. Such an environment has to have low- to high-level interfaces—from command-line tools to web interfaces. Following this paradigm, also online programming notebooks are an important offering in the scope of modern Science Gateways (using, e.g., JupyterHub[ii]).

In this contribution, we discuss a Science Gateway with rich backend systems and data stewardship support behind it, built for a community of researchers focusing on plant microbe interactions. The »Virtual Environment for Research Data and Analysis« (VERDA) for the Collaborative Research Centre »Genetic diversity shaping biotic interactions of plants (PlantMicrobe)« (CRC/TRR 356, funded by the German Research Council, DFG, project number 491090170) aims at supporting diverse research in different subfields of biology. The VERDA infrastructure will be jointly built by the comput-

https://dx.doi.org/10.15496/publikation-100323

ing centers of the participating universities (EKUT, LMU, and TUM), i.e. Zentrum für Datenverarbeitung (ZDV) in Tübingen and Leibniz Supercomputing Centre (LRZ) in Garching near Munich. A data stewardship component will be provided as well within VERDA, which is an effort led by Helmholtz Munich and TUM's MDSI (Munich Data Science Institute). The environment will support data-heavy and data-driven workflows not only from a computing-power, storage-space, and performance point of view, but also takes care of usability, FAIR (Findable, Accessible, Interoperable, Reusable)[iii] Research Data Management (RDM), and data curation. The implementation of well-controlled data sharing and publication mechanisms will be an essential point in supporting collaborative research within CRC/TRR 356. With a community-driven approach beyond CRC/TRR 356, VERDA aims at implementing data science and data management principles put forward by the German National Research Data Infrastructure (NFDI)[iv]. Thus, VERDA should help embed the TRR 356 in a broader context and optimally support biologists conducting almost twenty subprojects on investigating genetic aspects of biotic interactions between plants (e.g. with fungi). From symbiotic interactions of mycorrhizal fungi with trees to the problematic interplay of microbes and crops, understanding the cross-kingdom interactions between plants and microbes involves *omics, image analysis, and more. Targeting such a competitive field, VERDA will be of substantial importance for the success of the collaborative research agenda envisaged.

IT projects are crucial for efficient processing and modern Research data management in collaborative science. We aim to support practices and processes to manage, store, share, and preserve research data throughout its lifecycle. In biology, RDM faces challenges due to the large volumes of complex data generated by techniques like genomics, proteomics, metabolomics, and imaging. This influences design choices throughout the critical steps in the research data lifecycle, where we have to consider: i) the development of data management plans (DMPs); ii) Data storage and preservation in backup/archive systems, where large data volumes require specialized storage solutions; iii) the provisions (PIDs, metadata) to properly organize and document the data; iv) data sharing and publication; and v) long-term preservation and availability of critical datasets. All this helps to address the reproducibility crisis observed in biology[v] and other sciences. The aim is to make sure that data is discoverable, understandable, and reusable—as formalized in the FAIR principles. A strong collaboration with the NFDI, as indicated above, will help us to tackle domain-specific challenges with our infrastructure. NFDI establishes a sustainable intra- and interdisciplinary infrastructure framework for Research data management, aiming to provide researchers with the tools and services they need to manage, share, and preserve their research data securely and sustainably. NFDI consortia focus on coordination, harmonization, and interoperation of existing infrastructures. Thus, »implementation projects« producing special-purpose research environments and data management systems (such as VERDA) accordingly can perfectly complement the NFDI framework and contribute to its effectiveness. With VERDA, we strongly connect to and collaborate with the NFDI consortium DataPLANT[1], which stands for »Data management platform for high throughput technologies in plant research« and focuses on providing Research data management infrastructure and services for the community addressed.

Below, we present the frame and the concept for VERDA as an example of a modern Science Gateway architecture in more detail. Section II gives a glimpse into the requirements of the system, as derived from example use cases. Section III presents the basic building blocks we currently envisage for our environment, while Section IV lays out federation strategies for the geographically distributed multisite set-up we are planning. Section V concludes our short paper at hand and gives an outlook on further development and our plans for data stewardship.

2. USE CASE EXAMPLES AND REQUIREMENTS

Here, we will give a brief overview of the research topics that are covered by the institutions participating in the transregional collaborative research center and highlight the implied design necessities for the VERDA.

The research projects bundled in TRR 356 Plant-Microbe focus on beneficial and pathogenic interac-

[1] https://www.nfdi4plants.de

tions between plant hosts and microorganisms and the underlying molecular mechanisms determining the outcome of these interactions. The main research strategy is to record natural genetic variation in organisms involved in these interactions, using it to discover genetic determinants that shape the interactions and enhance understanding for optimizing symbiosis, pathogenic defense, and improving plant health.

Research projects of the TRR 356 PlantMicrobe can be roughly classified into the project areas ›Actors‹ and ›Mechanisms‹. The ›Actors‹ area aims to identify candidate genes by studying the natural diversity of plant-microbe interactions at the metabolome, genome, and transcriptome levels. The ›Mechanisms‹ area analyzes these candidates using various biochemical and imaging-based assays to understand how they impact plant health. As can be seen, by this abridged summary, experts from many different fields are involved in the transregional collaborative research center.

Another challenge caused by the data-driven approach of TRR 356 PlantMicrobe is the large amount of data that needs to be stored and analyzed. High-resolution microscopic images and *omics datasets can easily take up hundreds of gigabytes of storage space. The transregional nature of the program multiplies this challenge, because data needs to be accessible for research groups independent of their geographic location with acceptable access times for further bioinformatic analysis. Since the generation of these complex datasets requires a lot of resources, their quality and reusability must be ensured.

Therefore, to handle the large amounts of data generated by the projects and make it accessible for further analysis regardless of location, a robust data storage and management system is needed. Standardized formats with rich metadata will ensure the quality and reusability of the complex datasets. In addition, an efficient data analysis system is required to process and analyze the data.

3. BASIC COMPONENTS

In this section, we lay out the basic components of our system as we currently envisage it and plan—with necessary adaptations—to implement it in the coming years. The system will be a federated multi-site infrastructure, connecting the universities within CRC/TRR 356 and their computing centers (ZDV, LRZ—cf. Introduction). We first discuss the Research data management backend (Section III-A), then the cloud-computing backend (Sections III-B, III-C), and finally the user interfaces essential for a Science Gateway (Section III-D).

3.A. RDM SYSTEM

The RDM system in VERDA will be implemented following three principles: i) facilitating the management of datasets, from storage over metadata enrichment to assignment of unique identifiers within CRC/TRR 356 and beyond; ii) providing mechanisms (protocols, APIs, GUI) to efficiently exchange data between computing and storage systems (including custom lab devices and researcher laptops); iii) enabling rights management, data publication, and data archival. Principle ii), and even more so principle iii) require interfacing with the existing computing, storage, and archival facilities at the participating universities. To this purpose, the infrastructure parts of the IT subproject of CRC/TRR 356 are run by the computing centres of the participating universities, and a collaborative dialogue with the university libraries has already been started.

We will construct the system starting out simply with the allocation of storage growing from initially ~50 TB to over 200 TB with time. Then, managed data and metadata stores will be set up at each site, which shall be federated in the course of the project. One basic idea, following the techniques also applied in DataPLANT, is a distributed installation of GitLab[vi] covering the Munich and Tübingen sites. Such a technique has already been used by other large Research Data Management projects, e.g. in neurobiology[vii]. Clearly, FAIR data handling requires using GitLab in a uniform manner, where in particular the addition of metadata in a standardised subpath of each repository is mandatory. In our approach, metadata according to the ISA (investigation—survey—asset) model and ISA-TAB[viii] standard are added, after envisaged format checks, as files to the dataset, by which datasets shall ultimately be compliant with the packaging concept of an »Annotated Research Context« (ARC[ix], a profile implementation of RO-Crates[x]). In such a framework, metadata can be directly managed even on a laptop by a non-experienced researcher. GitLab commit hashes then can serve as the first sort of unique IDs for a version of a research dataset. GitLab commit histories together with the ISA metadata contain essential information to track the provenance of datasets within CRC/TRR 356. Once a dataset is published, we envisage the assignment of DOIs or other Handle.net-based[xi] globally unique PIDs. One challenge in this approach will certainly be the storage and transfer of huge files, which partially can be accommodated by Git's Large File Storage (LFS) feature[xii]. For the transfer from and to computing systems, we will provide easily usable mechanisms (e.g. python modules and/or a REST API), which will have to stand the test of benchmarks. As this research data management system with all its components is hosted close to ZDV's and LRZ's high performance file systems, which actually are also used for supercomputing, we are confident to provide competitive speed for analyses in a data-driven scientific environment, which demands analytics solutions similar as in the industrial »Big Data world«. Latencies in the access of often-used datasets will be avoided by automated dataset replication, mostly on data ingest. Whether all or only a part of them will be covered by the method will depend on the cost/benefit ratio and thus also on the contemporary development of inter-site network connectivity, mostly by third parties.

An important feature of GitLab as a basic system in VERDA is certainly the management of its fine-grained rights, which will be coupled to our global unified login and Identity & Access Management solution (cf. Section IV-A) for the CRC/TRR 356 IT. This will help researchers to share data and publish data exactly as they want it. Clearly, we advocate not only FAIR but also Open Data; yet, certain demands of researchers as embargoes or rights control have to be catered for. Our rights management and data publication systems will also be a cornerstone for CRC/TRR 356 researchers to fulfil the demands of the Nagoya protocol[xiii].

In its final stage, our vision is for the system to support researchers throughout their daily work with data. This includes storing hot/intermediate data and end results. Additionally, it should integrate existing databases, electronic lab notebooks (ELNs), and lab data management systems used in universities. The project aims to bring together the worlds of databases, ELNs, and FAIR Research data management, leveraging the experience of similar ongoing projects at participating computing centres. This will foster closer collaboration and address the problem in various project contexts.

Finally, we will provide rich, efficient, and easily usable interfaces for transferring data to the universities' long-term storage systems (archival systems of computing centres or libraries), and for publishing data. Thus, researchers can publish data via a variety of systems that are ideal for them or even strictly required to be used (e.g. institutional repositories, discipline-specific data repositories). In these processes, the ISA-TAB metadata files are kept with the data and updated, adding in particular back-references to the previous (or coexisting) storage locations within VERDA as important provenance information.

3.B. CLOUD-COMPUTING BACKEND: GENERAL DESIGN CONSIDERATIONS

A state-of-the-art computing and analysis platform will be implemented. On the one hand, this makes a powerful IaaS[2]/Cloud-Computing environment accessible on both sites (ZDV, LRZ) which can host permanent services (e.g. dataretrieval APIs) as well as intuitive user interface components. Correspondingly, VERDA includes financial support for managed virtual machines (managed VMs, »managed servers«) based on the de.NBI[xiv] Cloud infrastructure at ZDV and the VMWare-based managed-server infrastructure at LRZ[3]. These machines will be also available for real-time quick data analysis and similar tasks, possibly augmented by user-managed VMs on the LRZ Compute Cloud[4] or the de.NBI cloud. For more compute-heavy tasks we will facilitate data analytics (and computation in general) on High Performance Computing (HPC) resources already available to the researchers through their universities/institutes (LRZ: Linux Cluster[5], ZDV: BinAC[6]).

[2] We focus on using virtual machines for our workload; this workload is coming in containers (see also below) or as bare software with reproducible installation recipes; container orchestration is not explicitly envisaged.

[3] https://doku.lrz.de/display/PUBLIC/Managed+Server

[4] https://doku.lrz.de/display/PUBLIC/Compute+Cloud

[5] https://doku.lrz.de/display/PUBLIC/Linux+Cluster

[6] https://wiki.bwhpc.de/e/BinA

For an optimum interaction of these computing systems with the RDM System, interfaces have to be made available for computing processes acquiring and storing data. Ideally, addressing VERDA's RDM System should be accessible to computing jobs as easily as a normal file system. Where this cannot be achieved via »live« interfacing (e.g. by exposing storage REST APIs following the S3 standard, and addressing them[xv]), convenient staging mechanisms have to make the files available on file systems directly used by the computing machines. Such staging mechanisms enable easy dataset transfer between cloud-computing/HPC environments in our ecosystem, leveraging previous developments[xvi].

We envisage larger parts of the computations to be executed interactively, which makes JupyterHub (cf. Section III-D below) an important component that will be utilized. This should be backed by a maximum degree of automatization in backend usage. I.e. details should be masked as far as possible by appropriate automatization and orchestration, which will be a challenge in our complex computing ecosystem. Yet, advanced users shall be enabled to easily access useful data about the actual execution technique and target system (e.g. in order to launch a correct number of parallel analysis processes). On the workflow level, automatization can and will be reached by employing state-of-the-art and community-standard tools, where we have in particular Galaxy[xvii] in mind, but may also consider Nextflow[xviii] and Snakemake[xix]. For a homogeneous representation of different workflow engines, CWL (Common Workflow Language) will be used.

3.C. CLOUD-COMPUTING BACKEND: TÜBINGEN REFERENCE IMPLEMENTATION

With EKUT/ZDV leading the Cloud-Computing component of VERDA, a reference computing-backend implementation for VERDA has been initiated at EKUT. The Tübingen-based de.NBI Cloud site, employed for the PlantMicrobe project, boasts of geo-redundant server locations within Tübingen, extensive expertise in secure data processing and storage, and access to high performance GPU servers. The presence of supportive staff ensures productive and transparent collaboration with the cloud.

After the initiation of VERDA, the de.NBI Cloud site in Tübingen provides a range of VMs and network infrastructure to support the Plant Microbe project (Figure 1).

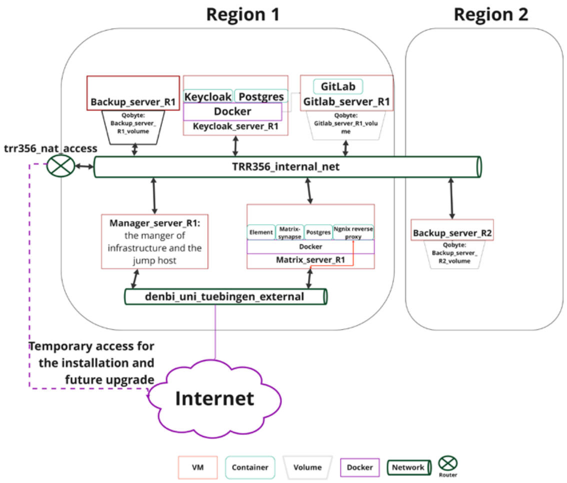


Figure 1: General overview of the CRC/TRR 356 components in de.NBI Cloud infrastructure in Tübingen (state 04/2023).

The set-up of our entire infrastructure relies on an infrastructure-as-code/automatized-deployment approach based on the Ansible open-source framework[xx]. With Ansible and its easy declarative language, we can automate tasks such as provisioning, configuration, and deployment of virtual machines. This enables us to focus on optimum systems for research rather than the complexities of IT infrastructure management. Also, it facilitates the collaborative set-up of infrastructure by sharing Ansible scripts. At Tübingen, a managing and jump-host VM (Manager server R1 in Figure 1) is used to run Ansible scripts for setting up the other project VMs.

Two other notable virtual machines (cf. Figure 1) that are essential for central functionalities of VERDA (cf. Sections III-D, IV-A) are the Matrix server R1 VM and Keycloak server R1 VM. Both virtual machines use Docker containers for all components, providing a lightweight and portable way to package and deploy applications. The Matrix server R1 VM is designed to host the Matrix.org ecosystem, including the Element chat client, the Matrix/Synapse server, the PostgreSQL database, and Nginx reverse proxy. The Keycloak server R1 VM provides a secure and centralized authentication server for the project. This approach provides the team with a secure, scalable, and easy-to-manage authentication system for their project.

The Gitlab server R1 VM is pre-configured with all the necessary components for GitLab, including the operating system, required packages, and dependencies. It is an essential backend to VERDA's Research Data Management system (see Section III-A).

Finally, the Backup server R1 and Backup server R2 (Figure 1, upper left and lower right parts) play a crucial role in ensuring the safety and security of the data in the PlantMicrobe project. These servers actually back up the other virtual machines in both regions of the de.NBI Tübingen site. In case of any failure or disaster, the backup virtual machines can quickly restore the data to the affected virtual machines, minimizing the impact on the project.

In the de.NBI Cloud site in Tübingen, the Plant Microbe/VERDA project uses two networks to connect its virtual machines: an internal network called »TRR356 internal net« and an external network called »denbi uni tuebingen external.« The »TRR356 internal net« is an internal, isolated (secure and private) network for virtual machines (e.g. Manager server R1, Matrix server R1, Gitlab server R1, Keycloak server R1, Backup server R1, and Backup server R2). On the other hand, the »denbi uni tuebingen external« network is an external network that provides access to the internet for machines where this is necessary (Manager server R1 and Matrix server R1). The minimization of Internet access is reducing the risk of security breaches.

Where significant amounts of secure and high-performance storage are needed (in particular Backup server R1 and Backup server R2), the Tübingen site relies on a Quobyte[xxi] system. The system offers advanced security mechanisms and data management features, such as erasure coding and data compression.

3.D. USER INTERFACE MODULES

As the first user interface components within our environment, we have actually been asked to implement communication solutions for the CRC/TRR 356. Hence, we have set up a system that can provide group chats, video conferencing, file sharing, and more. Our system is based on Matrix[xxii], an open-source communication protocol that is designed for secure, decentralized communication. At its core, it has the open-source Matrix homeserver »Synapse«, written and maintained by the Matrix.org Foundation. Element[xxiii], a Matrix-based end-to-end encrypted messenger and secure collaboration app, is recommended to all CRC/TRR 356 members in order to access the chat groups of the collaboration. This communication system is an essential component of the project, as it will allow the members to collaborate, share information, and make important decisions in real-time.

In the coming months and years, we will make essential steps to build a full-fledged Science Gateway / VRE upon our backend systems. We will couple data-management user interfaces (e.g. laboratory notebooks and microscopy software) to our environment as well as web-based programming interfaces. In particular, we envisage offering online programming notebooks via JupyterHub[xxiv]. These notebooks will be tightly integrated with our research data management and computing systems, offering a unified and very flexible data-analysis environment to the users.

To achieve this level of integration, the VRE will be designed to interface with RDM and e-Infrastructure in a secure, efficient, and scalable way. This will involve implementing APIs and/or web services to facilitate communication between systems.

4. SINGLE SIGN ON AND FEDERATED ARCHITECTURE

4.A. SINGLE SIGN ON VIA LIFE SCIENCE AAI AND KEYCLOAK

In a multi-component environment such as VERDA, it is essential to provide users with a unified log-on mechanism (so-called Single Sign On/SSO) and an easy way to obtain an account. When making accounts and verifying identities, we thus rely on the internationally established Life Science Authentication and Authorization Infrastructure (AAI) put forward by the European research infrastructure collaboration ELIXIR[xxv].

When joining our collaborative environment, users sign up with the Life Science/ELIXIR AAI as an Identity Provider (IDP). Users from all eduGAIN institutions, including those active in the CRC/TRR 356, can easily sign up via delegated verification at their home identity provider (local computing center). Additionally, »social« logins like ORCID, Google, or LinkedIn can be used for authentication. The Elixir ID, a globally unique and unchanging identifier, serves as the primary source of identity in the Elixir AAI and can be used as a reference by us.

Using an established, unified infrastructure for authentication is essential. To enhance authorization flexibility, we integrate this approach with Keycloak[xxvi], a widely adopted open-source tool for identity and access management. Keycloak allows integration with existing identity providers, supporting authentication and authorization through protocols like OAuth 2, OpenID Connect, and SAML 2.0. It acts as a centralized SSO provider for VERDA services, facilitating seamless integration with various services in a heterogeneous landscape. By using roles, attributes, or custom logic, we can then define fine-grained access controls. This approach enables us to handle various service requirements and restrict information exposure.

Although Keycloak provides technical means, mapping between third-party systems is still required, as well as specialized workflows for specific requirements like DOI registration to ensure data quality. Keycloak serves as a common link between services to manage SSO and authorization.

4.B. NETWORK AND TRAFFIC ENCRYPTION

Core backend systems at both sites (ZDV, LRZ) will be in secure network segments (cf. Section III-C), where however the integration of VERDA with production infrastructure at the computing centers (storage, virtual machines, ...) may necessitate some shared usage of subnetworks. Nonetheless, we aim to place all of the permanent/core virtual machines of VERDA at each site in a separate network segment which can be more easily firewalled. Finally, also VPN-based connectivity between these subnetworks will be considered. While this has been realized in previous projects VERDA partners participated in, for example, securing unencrypted traffic adds complications and may affect reliability if the VPN connection breaks down[xxvii]. Independent of the decision on that point, all traffic within the VERDA system will be encrypted (as supported by default in most modern IT systems), thus fulfilling an important basic security requirement.

4.C. FEDERATION CONCEPTS FOR DATA AND COMPUTE

For the federation of data, GitLab will be installed on both sites (LRZ and ZDV). Currently, two different options for establishing a two-site GitLab are being investigated. The first solution involves implementing a multi-node GitLab with three nodes[xxviii]. Two would reside at ZDV, while one would reside at LRZ. Because the complexity of this solution is high concerning the project size, another option would be to set up a single-node GitLab which runs on the ZDV infrastructure with regular backups on the LRZ.

Computing federation concepts have yet to be fully explored. Because LRZ and ZDV have separate infrastructures, dividing tasks to use computing power from both sites at the same time would be difficult. As a result, it is sensible to orchestrate tasks on a higher level and assign a singular task to one site while assigning them using a load balancer so that both sites share computing power equally.

5. CONCLUSIONS

This contribution has laid out the characteristics of a »Virtual Environment for Research Data and Analysis« (VERDA) we are building for a project of researchers focused on plant microbe interactions. We try to implement the FAIR principles and systematic, access-rights-aware data management as well as highly performant and well-usable data analytics workflows. To this end, VERDA provides an RDM System, and a computing environment, and makes use of state-of-the-art federation and Single Sign On approaches.

We are confident that this ambitious project will be a success in supporting our project scientists within the CRC/TRR 356 »Genetic diversity shaping biotic interactions of plants (PlantMicrobe)« and beyond. Learning from earlier experience, we employ a substantially more flexible concept of a Science Gateway, and we introduce the infrastructure to scientists with a very strong data stewardship approach. In the VERDA set-up phase, we have allocated over one-third of the workforce to data stewardship, training, and support actions. This exceptional investment shall make sure that all subprojects of the CRC/TRR 356 can understand and leverage the benefits of professional-grade computing resources and modern approaches to data management including the FAIR principles. It shall also foster homogeneity in the usage of data formats across all subprojects and ensure that all subprojects follow our ISA-based metadata concept.

The system, which we will forge following the principles developed within the NFDI initiative and in particular in the consortium DataPLANT, shall thus be a prime example of successfully implemented Research data management among Germany's bioscience collaborations. We will publish our concepts as they evolve and expect them to serve as a blueprint for similar projects. Furthermore, we will prepare our systems to be sustained beyond the project lifecycle, by embedding them from the beginning in larger-scale efforts such as NFDI and avoiding parallel work.

6. ACKNOWLEDGMENT

Parts of this work and its authors have been funded by DFG through the CRC/TRR 356 »PlantMicrobe« grant no. 491090170. We acknowledge the support of DataPLANT (NFDI 7/1—442077441) as part of the German National Research Data Infrastructure. Further parts of the work presented in this paper has been supported by the SDC BioDATEN which is supported by the Ministry for Science, Research, and Arts of the federal state of Baden-Württemberg. We also acknowledge the use of the de.NBI Cloud and the support by the Federal Ministry of Education and Research (BMBF) through grant no 031A535A.

7. REFERENCES

i K. A. Lawrence and N. Wilkins-Diehr, »Roadmaps, not blueprints: paving the way to science gateway success,« in *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond, XSEDE '12*, C. Stewart, Ed. July 2012, Art. no. 8. https://doi.org/10.1145/2335755.2335837

ii A. Zonca and R. S. Sinkovits, »Deploying Jupyter Notebooks at scale on XSEDE resources for Science Gateways and workshops,« *PEARC '18: Proceedings of the Practice and Experience on Advanced Research Computing: Seamless Creativity*, July 2018, Art. no. 8. https://doi.org/10.1145/3219104.3219122

iii M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J-W. Boiten, L. Bonino da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Corsas, I. Dillo, O. Dumon, S. Edmonds, C. T. Evolo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Growth, C. Goble, J.S. Grethe, … and B. Mons, »The FAIR Guiding Principles for scientific data management and stewardship,« *Sci. Data*, vol. 3, Art. no. 160018, 2016. https://doi.org/10.1038/sdata.2016.18

iv Nationale Forschungsdateninfrastruktur (NFDI) e.V. »NFDI | Nationale Forschungsdateninfrastruktur e.V.,« 2021. Accessed: Mar. 19, 2023. [Online.] Available: https://www.nfdi.de

v M. Baker, »1,500 scientists lift the lid on reproducibility,« *Nature*, vol. 533, no. 7604, pp. 452–454, 2016. https://doi.org/10.1038/533452a

vi GitLab B.V. »The DevSecOps PLatform | GitLab.,« 2023. Accessed: Mar. 19, 2023. [Online.] Available: https://www.gitlab.com

vii G-Node collaboration. »About—G-Node GIN,« 2023. Accessed: Mar. 19, 2023. [Online.] Available: https://gin.g-node.org/G/Node/Info/wiki/about

viii P. Rocca-Serra, M. Brandizi, E. Maguire, N. Sklyar, C. Taylor, K. Begley, D. Field, S. Harris, W. Hide, O. Hofmann, S. Neumann, P. Sterk, W. Tong, and S-A. Sansone, »ISA software suite: supporting standards-compliant experimental annotation and enabling curation at the community level,« *Bioinformatics*, vol. 26, no. 18, pp. 2354–2356, 2010. https://doi.org/10.1093/bioinformatics/btq415

ix B. Venn, K. Schneider, K. Frey, H. L. Weil, J. Werner, F. Wannenmacher, T. Zajac, D. von Suchodoletz, B. Usadel, J. Krüger, C.

Garth, and T. Mühlhaus, »Fostering the democratization of research data by using the Annotated Research Context (ARC) as practical implementation,« presented at E-Science-Tage 2021: Share Your Research Data, 2021, doi: https://dx.doi.org/10.11588/heidok.00029769

x S. Soiland-Reyes, P. Sefton, M. Crosas, L. J. Castro, … and C. Goble, »Packaging research artefacts with RO-Crate,« *Data Science*, vol. 5, no. 2, pp. 97–138, 2022. https://doi.org/10.3233/ds-210053

xi L. Lannom, B. P. Boesch, and S. Sun, »Handle System Overview,« *RFC*, no. 3650, Nov. 2003. https://doi.org/10.17487/RFC3650

xii GitLab B.V. »Git Large File Storage (LFS) | GitLab,« 2023. Accessed: Mar. 19, 2023. [Online.] Available: https://docs.gitlab.com/ee/topics/git/lfs/

xiii E. Morgera, E. Tsioumani, and M. Buck, »*Unraveling the Nagoya Protocol: A Commentary on the Nagoya Protocol on Access and Benefit-sharing to the Convention on Biological Diversity,*« Leiden, Netherlands: Brill, 2015, pp. 444. https://doi.org/10.1163/9789004217188

xiv A. Tauch and A. Al-Dilaimi, »Bioinformatics in Germany: toward a national-level infrastructure,« *Briefings in Bioinformatics*, vol. 20, no. 2, pp. 370–374, 2019.

xv MinIO Inc. »MinIO | High Performance, Kubernetes Native Object Storage,« 2023. Accessed: Mar. 19, 2023. [Online.] Available: https://min.io

xvi J. Munke, M. Hayek, M. Golasowski, R. J. García-Hernández, F. Donnat, C. Koch-Hofer, P. Couvee, S. Hachinger, and J. Martinovič, »Data System and Data Management in a Federation of HPC/Cloud Centers,« in *HPC, Big Data, and AI Convergence Towards Exascale*, O. Terzo and J. Martinovič, Eds., Boca Raton (FL): CRC Press, 2022, chap. 4, pp. 59–77. [Online.] Available: https://dx.doi.org/10.1201/9781003176664-4

xvii Galaxy Community, »The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update,« *Nucleic Acids Research*, vol. 50, no. W1, pp. W345–W351, 2022. https://doi.org/10.1093/nar/gkac247

xviii P. Di Tommaso, M. Chatzou, E. W. Floden, P. Prieto Barja, E. Palumbo, and C. Notredamme, »Nextflow enables reproducible computational workflows,« Nature Biotechnology, vol. 35, no. 4, pp. 316–319, 2017. https://doi.org/10.1038/nbt.3820

xix J. Koster and S. Rahmann, »Snakemake—a scalable bioinformatics workflow engine,« *Bioinformatics*, vol. 28, no. 19, pp. 2520–2522, 2012. https://doi.org/10.1093/bioinformatics/bts480

xx RedHat Inc. »Ansible Documentation,« 2020. Accessed: Mar. 19, 2023. [Online.] Available: https://docs.ansible.com/

xxi Quobyte Inc. »100% Software Storage—Quobyte,« 2023. Accessed: Apr. 2, 2023. [Online.] Available: https://www.quobyte.com/

xxii Matrix.org Foundation and Collaborators. »Matrix.org,« 2023. Accessed: Apr. 2, 2023. [Online.] Available: https://matrix.org/

xxiii Element collaboration, New Vector Ltd. »Element | Secure collaboration and messaging,« 2023. Accessed: Apr. 2, 2023. [Online.] Available: https://element.io/

xxiv A. Zonca and R. S. Sinkovits, *PEARC '18: Proceedings of the Practice and Experience on Advanced Research Computing: Seamless Creativity*, July 2018, Art. no. 8.

xxv ELIXIR. »Overview of the AAI | ELIXIR,« 2020. Accessed: Apr. 2, 2023. [Online.] Available: https://elixir-europe.org/platforms/compute/aai/overview

xxvi JBoss (Red Hat Inc.), Keycloak Community. »Keycloak,« 2020. Accessed: Nov. 6, 2020. [Online.] Available: https://www.keycloak.org/

xxvii F. Donnat, »LEXIS: Large-scale Execution for Industry & Society Deliverable D4.5. Definition of mechanisms for securing federated infrastructures,« LEXIS Project (H2020 GA No. 825532), Tech. Rep., Mar. 31, 2020. Accessed: Mar. 19, 2023. [Online.] Available: https://cordis.europa.eu/project/id/825532/results

xxviii GitLab B.V. »Reference architecture: up to 3,000 users | GitLab,« 2023. Accessed: Feb. 22, 2023. [Online.] Available: https://docs.gitlab.com/ee/administration/reference architectures/3kusers.html

AN APPROACH TO PROVIDE SERVERLESS SCIENTIFIC PIPELINES WITHIN THE CONTEXT OF SKA

CARLOS RÍOS-MONJE, MANUEL PARRA-ROYÓN*, JAVIER MOLDÓN, SUSANA SÁNCHEZ-EXPÓSITO,
JULIÁN GARRIDO, LAURA DARRIBA, M. ANGELES MENDOZA, JESÚS SÁNCHEZ,
LOURDES VERDES-MONTENEGRO
Instituto de Astrofísica de Andalucía, Gta. de la Astronomía, s/n, 18008 GRANADA, SPAIN
*mparra@iaa.es

JESÚS SALGADO
SKA Observatory, Jodrell Bank, Macclesfield SK11 9FT, UK

ABSTRACT

Function-as-a-Service (FaaS) is a type of serverless computing that allows developers to write and deploy code as individual functions, which can be triggered by specific events or requests. FaaS platforms automatically manage the underlying infrastructure, scaling it up or down as needed, being highly scalable, cost-effective and offering a high level of abstraction. Prototypes being developed within the SKA Regional Center Network (SRCNet) are exploring models for data distribution, software delivery and distributed computing with the goal of moving and executing computation to where the data is. Since SKA will be the largest data producer on the planet, it will be necessary to distribute this massive volume of data to the SRCNet nodes that will serve as a hub for computing and analysis operations on the closest data. Within this context, in this work we want to validate the feasibility of designing and deploying functions and applications commonly used in radio interferometry workflows within a FaaS platform to demonstrate the value of this computing model as an alternative to explore for data processing in the distributed nodes of the SRCNet. We have analysed several FaaS platforms and successfully deployed one of them, where we have imported several functions using two different methods: microfunctions from the CASA framework, which are written in Python code, and highly specific native applications like wsclean. Therefore, we have designed a simple catalogue that can be easily scaled to provide all the key features of FaaS in highly distributed environments using orchestrators, as well as having the ability to integrate them with workflows or APIs. This paper contributes to the ongoing discussion of the potential of FaaS models for scientific data processing, particularly in the context of large-scale, distributed projects such as SKA.

Keywords: *Function as a Service; Serverless; HPC; Cloud Computing; SKA; Radio Astronomy; Open Science*

1. INTRODUCTION

The evolution of computing paradigms has been driven by the need to process ever-increasing amounts of data and perform complex computations more quickly and efficiently, benefiting many scientific fields. With the advent of personal computers, client-server computing emerged and scientific applications were divided into two parts: the client (user access and interface) and the server (back-end processing) where the client would request data from the server, which would process the request and return the data to the user.

As computational and storage requirements increased, distributed processing emerged as a way to leverage the processing power of multiple computers or clusters connected via a network allowing for faster processing of large datasets and complex computations in many scientific fields. Grid computing was one of the most successful models to provide a distributed computing environment at the time but it was also a complex system. High Performance Computing uses supercomputers with large volumes of memory and specialised architectures where hardware is highly optimised for certain processes, to perform complex calculations and simulations and it is used in scientific research, engineering and other fields where large amounts of data need to be processed extremely quickly. In the early 2000s until today, as a way to provide on-demand access to

https://dx.doi.org/10.15496/publikation-100323

computing resources over the internet, Cloud Computing (CC) as computing paradigm allows users to get computing resources (such as Virtual Machines [VMs], storage, networks or software as services) on a pay-on-demand basis.

CC has brought a revolution in the way computing, storage and other resources are managed, giving institutions, businesses and individuals the ability to harness the potential in the form of very different on-demand services. CC provides flexible access to computing resources that can be scaled up or down as needed, allowing researchers to process and analyse large amounts of data efficiently as well as to provide access to specialised computing resources such as GPUs and FPGAs that can accelerate scientific pipelines. CC makes it easier for researchers to collaborate, sharing data or the execution environment, and to publish and share their work through cloud-based services, being digital infrastructures a key player for scientific progress and Open Science. Within CC, a paradigm called Serverless[i] or Function-as-a-Service (FaaS) has emerged in recent years, which allows code to be executed (as functions) without having to manage[ii] or provision VMs, computing power or storage resources. Serverless allows you to focus on writing and deploying code or functions, being this is a natural extension of CC, as it provides even more flexibility and scalability than traditional CC by allowing developers to split their applications/services into smaller, more manageable functions that can run independently and in a more streamlined way.

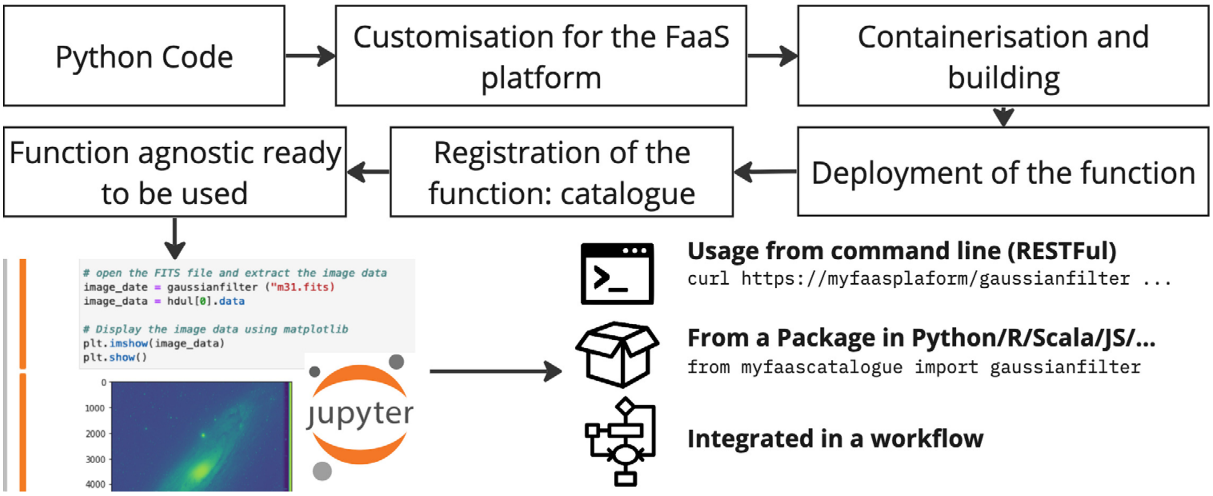
The Square Kilometre Array (SKA) is a new generation radio telescope currently under construction and will be one of the most advanced and powerful telescopes in the world, capable of observing the sky with an unprecedented level of detail and sensitivity. It will deliver about 700 PB of data products per year. Given these attributes, the SKA will require significant computing resources for both data processing and analysis. The SKA data will be delivered to a Global Network of SKA Regional Centres (SRCNet) which will provide access for an international community to SKA Observatory data and the analysis tools as well as the processing power to fully exploit their science potential, so SRCs is where the SKA science will be done. It will require the use of advanced computing technologies and techniques, such as HPC, distributed CC and Machine Learning (ML).

In this context, we believe that Serverless computing can be highly beneficial to SRCNet data processing capabilities due to its scalability, cost-effectiveness, efficiency and reliability. With Serverless, researchers can scale their computing resources as needed and ensure that SKA data processing workflows are highly available across different global regions, as well as they can design functions that can be integrated from anywhere, such as Jupyter Notebooks, workflows or command line, among others. In this paper we explore the feasibility of using Serverless computing for the design of several functions commonly used in astronomy pipelines and in particular of some of the operations involved in the data workflows of SKA precursors telescopes (such as MeerKAT, ASKAP or MWA) and to tackle image analysis, data cube visualisation and analysis, spectral analysis, source extraction, among others.

As for the content, in the section 2 we will review the application of this Serverless paradigm to scientific problems, then in section 3 the different solutions available to deal with the model from the point of view of architecture, services and composition will be addressed as well as we will detail the serverless and FaaS approach that has been chosen including a subset of radio interferometry functions that have been implemented. Finally, in section 4 we address the conclusions and the future work that we propose.

2. BACKGROUND

As commented, Serverless computing is a popular CC model that allows developers to run code and applications without worrying about underlying infrastructure. This model provides scalability and cost-effectiveness by automatically handling the provisioning and management of servers, scaling resources based on demand, and handling fault tolerance and availability. Cloud providers, such as Amazon Web Services, Microsoft Azure, and Google Cloud, offer serverless computing services that provide a wide range of capabilities[iii], from simple function execution to complex event-driven architectures, and integrate with other cloud services like storage, network or more recently the use of ML or Artificial Intelligence (AI) capabilities as a service[iv].


Python Code
Customisation for the FaaS platform
Containerisation and building
Deployment of the function
Registration of the function: catalogue
Function agnostic ready to be used
0
1000
2000
3000
4000
jupyter
Usage from command line (RESTFul)
From a Package in Python/R/Scala/JS/...
Integrated in a workflow

```
# open the FITS file and extract the image data
image_date = gaussianfilter ("m31.fits)
image_data = hdul[0].data

# Display the image data using matplotlib
plt.imshow(image_data)
plt.show()
```

```
curl https://myfaasplaform/gaussianfilter ...
```

```
from myfaascatalogue import gaussianfilter
```

Figure 1: Steps from a simple python code to a function in a FaaS platform

Serverless computing is being adopted across a wide range of industries[v] and use cases, from web and mobile applications to Big Data processing and scientific computing. However, there are still challenges to overcome, such as the lack of standardised development and deployment practices, limited tooling and debugging capabilities, and issues related to security and compliance. Despite these challenges, the state-of-the-art in serverless computing is rapidly evolving, with new features and services being added to cloud provider offerings[vi], and the adoption of serverless computing increasing among developers and organisations.

In the CC context, the current serverless landscape was introduced during an Amazon Web Services 'Reinvent' event in 2014, and since then, multiple cloud providers and industrial and academic institutions have introduced their own serverless platforms. Serverless platforms are a natural step after VMs and container technologies, where each stage led to lighter computing units in terms of resource consumption, cost and development speed. Other CC providers followed in 2016 with the introduction of Google Cloud Functions, Microsoft Azure Functions and IBM OpenWhisk, the latter released as Open Source (OS) to the community. In addition to public cloud providers that support FaaS, there are several initiatives to bring Serverless to private environments from OS developments ready to be deployed in our own infrastructures, such as OpenFaaS or OpenLambda, among others. FaaS has been enriched with high-performance, lightweight fast polyglot runtimes, and Ahead-of-Time (AOT) compilation features for a faster startup times and lower memory usage, as proposed with GraalVM[1].

Within the scientific environment there are several research works on the application of the serverless model to workflows in different areas of knowledge. Authors[vii] look into the problem of scheduling scientific workflows and discussing challenges related to workflow scheduling with Cloud functions. Other studies[viii] address FaaS execution systems with a hybrid solution for scientific workflows and from another point of view in[ix], challenges in serverless edge computing and open research opportunities for ML applications are depicted. Thus, within the field of astrophysics, the serverless model has still not been sufficiently tackled. In this paper we want to initially explore how FaaS can be an option for the deployment and execution of scientific workflows in this field of knowledge.

3. SERVERLESS COMPUTING APPROACH FOR WORKFLOWS IN ASTRONOMY

Serverless computing is typically implemented using a Function-as-a-Service model. With FaaS, developers and scientists write small, self-contained functions that perform a specific task or respond to a particular event. These functions are deployed to a serverless platform (such as AWS Lambda, Google Cloud Functions, or OS FaaS platforms in your own infra-

[1] https://www.graalvm.org/latest/docs/introduction/

structures) and are triggered by events, such as HTTP requests, database updates, or messages from a queue collector. When a function is triggered, the system will automatically allocate the resources regardless of the underlying platform, either HPC, CC, etc. to run the function. In this case the function runs in a stateless container, meaning that it does not store any persistent state or data between invocations. Once the function completes its task, the container is terminated, and the resources are released back to the platform.

In this scenario, if we want, for example, to create a function that takes an image, applies a Gaussian blur filter to smooth the image and returns the image data, we just need to write the code as usual and add some handlers for the input and output data of the function as shown in code listing 1. Once this is done, it will be necessary to build the function using a language such as Python, Go, Scala, NodeJS, or others. This code is then included into a container that provides the core component to provide the FaaS integration logic. The containerised function must then be registered in the catalogue within the FaaS platform so that it is available to be invoked from HTTP or from an API. Finally, the function can be called from virtually any application or service as it is published from a RESTful API or GraphQL, for example. With this publishing model the function could be used from a Jupyter Notebook for example as shown in Figure 1.

```
import cv2
import numpy as np
import astropy.io.fits as fits
from flask import Flask, request, Response
def main():
   # get the file from the request
   file = request.files['file']
   # read the FITS image
   hdul = fits.open(file)
   image = hdul[0].data
   # Gaussian filter:kernel 3x3, sigma 1.5
   fi = cv2.GaussianBlur(image, (3, 3), 1.5)
   # write FITS file
   hdul[0].data = fi
   hdul.writeto(file, overwrite=True)
   # return the filtered FITS
   return Response(open(file, 'rb').read(),
      mimetype='application/fits')
```

Code Listing 1: Code listing. 1: Example of an image filtering for a FaaS deployment.

Public serverless computing platforms, such as AWS Lambda or Google Cloud Functions, are offered as fully managed services by CC providers, meaning that the it takes care of all aspects of managing and scaling the platform, including underlying infrastructure, security, and maintenance as well as to provide a convenient and scalable way for developers to build and deploy serverless applications without having to manage the underlying infrastructure and paying-as-you-go. On the other hand, OS serverless computing platforms are community-driven projects that provide an alternative to commercial services. These platforms are often built on top of container orchestration systems like Kubernetes or Docker and allow developers to run serverless functions on their own infrastructure or in a public cloud.

Open Science has been gaining momentum in recent years, and many research organisations are looking for ways to adopt it into their workflows. One of the key components of Open Science is the ability to share research data and code openly and transparently. This is where OS platforms for serverless computing and Function-as-a-Service (FaaS) capabilities can be of great value. In order to make it easy for research organisations to adopt these platforms, it is important to provide here not only the code to run the platform and create the containers, but also the logic and software for the functions implemented on the platform. This ensures that researchers have a deep understanding of what the functions are doing internally, and can customise them to fit their specific needs, leading to more efficient and effective research. In this paper we have provided a repository (see section 3) with all these components to promote Open Science and making it easier for research organisations to adopt FaaS capabilities.

In Table 1, we provide a summary of these platforms that we have been working with to develop the set of functions detailed within the next subsection 3.A.

3.A. SERVERLESS WITH AN ASTRONOMY WORKFLOW

Within the context of working with and processing data from both SKA precursors and pathfinders telescopes[2] as well as other radio telescopes, the analysis

[2] https://www.skao.int/en/explore/precursors-pathfinders

Platform	State of development	Difficulty of installation	Cluster	Supported Languages
OpenFaaS	Active	Easy	Kubernetes or OpenShift	Go, Python, Node.js, Ruby, Java, C#, Dockerfile, etc
Knative	Active	Moderate	Kubernetes	Node.js, Python, Go, Quarkus, etc
Kubeles	Not actively maintained	Easy	Kubernetes	Go, Python, NodeJS, Ruby, etc.
Fission	Active	Easy	Kubernetes or OpenShift	Node.js, Python, Ruby, Go, PHP, Bash and Dockerfile
OpenWhisk	Active	Moderate	Kubernetes	Node.js, Python, Swift, Java and more

Table 1: Open Source FaaS Platforms

of interferometric data involves the concatenation of a series of steps, generally in linear way, but which can be approached from different points of view, being organised in a logical chain or workflow with/without dependencies. The steps broadly include data manipulation, where raw data from the radio telescope are managed to ensure that they are in a suitable format for analysis and all necessary conversions are performed. This is followed by flagging, which involves the removal of incorrect or corrupted data or data affected by noise or other artefacts, to ensure correct calibration and imaging of the data can be obtained. The next step would be calibration, which aims to generate calibration tables and apply them to the data to correct for instrumental and atmospheric effects. Usually, the data of the calibrated target source is split and averaged if needed. And finally, the imaging phase and self-calibration, where deconvolution, cleaning, imaging and residual calibration is performed for a final product of images or data cubes. Note that some instruments or observing modes will require variations of this general workflow, in particular with additional steps and verification functions (see an overall workflow in a paper on continuum imagine pipelines[x]).

To perform these steps, several astronomical tools can be used. The most widely used library package for interferometric data analysis is CASA[xi], currently implemented as Python modules, so it can be run interactively or via scripts or workflows. There are also several Python-based pipelines for different instruments, such as VLA CASA calibration pipeline, the e-MERLN CASA pipeline[xii], CARACAL[xiii], among others, which include tools and containers customised with the pipeline operations. Additionally, there are different standalone tools (binary executables) such as wsclean[xiv] (for imaging), aoflagger[xv] (flagging) that are specialised in specific aspects of the pipeline and can be much more efficient than CASA on specific tasks, including support for GPUs in some cases. In this way each step from the beginning of the raw data to the presentation of the image is a succession of steps where the results of each step are the inputs to the next ones, so that the processing can be highly automated and if possible distributed and scaled.

Under this view the workflow would typically involve the development of individual functions or microservices that perform specific tasks, such as data manipulation, cleaning, flagging, calibration or/and image processing. These functions would then be orchestrated together, selecting the most appropriate tool for each relevant combination of instrument and scientific objective, to form the overall serverless application by using a FaaS platform. This model offers advantages in terms of high availability, scalability and interoperability of tasks in a highly distributed environment such as the SRCNet, with globally distributed centres providing different computing platforms and diverse hardware, as well as configuring a SRCNet datamesh model[3]. FaaS would use an »execution planner« service to determine the optimal location for executing the function based on the location of the data, available resources, and cost. The function would then be deployed to the appropriate data centre, minimising large data transfers and latency and maximising performance.

In a context, from a simple FaaS model, functions are called on demand with no a priori scheduling on where or how they will be executed, as they are abstracted for the user, and it is the underlying container

[3] https://www.datamesh-architecture.com/

orchestrator that manages it with its policies. But in a complex and highly distributed environment such as a datamesh, it is necessary to operate the functions with an execution planner at the orchestrator level that transfers/decides the execution/computation of these functions to the specific resources where they are best suited. FaaS can help this model from the orchestrator by analysing the data and determining which functions should run on which parts of the data, distributing the computational workload across the distributed network to make decisions based on factors such as data availability, network latency and cost, ensuring that functions run in the most efficient and cost-effective way.

3.B. A TESTBED PLATFORM FOR SERVERLESS WITH FISSION

We have studied and tested Fission (see Table 1) as an OS platform that could be applied to scientific analysis to produce SKA advanced data products within the SRCNet platform. Fission has become more popular in the recent times due to aspects such as ease of use, implementation and deployment, integration with Kubernetes natively and scaling of functions using Kubernetes metrics, as well as the need to reduce the latency of functions in highly distributed environments, providing a fast cold start time of functions.

For this testbed we have deployed Fission on the Spanish Prototype of an SKA Regional Centre, the SPSRC[xvi], located in Granada, Spain, under a two-node Kubernetes cluster (v.1.24.0, using Rancher as orchestrator and Fission—release 1.6.0) and exposing the functions locally to the SPSRC projects. The installation steps can be found in the next section.

3.C. DEVELOPMENT AND DEPLOYMENT OF FUNCTIONS

Based on the radio interferometry workflow, we have selected a subset of functions that can be easily exported within the FaaS Fission platform. To show the adaptability of FaaS to practically any code/engineering or application that the user wants to execute, we will show as an example how to design a function for the imaging component tclean from CASA framework and as well the wsclean application. With these two possibilities, Python CASA code and binary application, we can highlight the capabilities of FaaS to be extended to any kind of software, as they can be easily containerised.

For reasons of space, all the details of installation, environments, execution and parameterisation are available from the project repository[4].

The first step towards creating a function in Fission is to define the environment where the function works. This environment will need all the necessary dependencies and packages, in addition to the FaaS Fission core components. We can create a custom environment from one of the pre-existing Fission environments[5]. To use the CASA framework, we will need to customise one Python environment with the corresponding Python packages. For the first approach, starting from a Python environment, we modify the Dockerfile, as indicated in code listing 2, to choose a proper base image, python:3.8-buster and then, we add the CASA packages, casatasks, casatools and casadata, to the requirements.txt file. In code listing 2 the last two lines include the basic services for FaaS to run on the platform. Then, we build the image and push it to a public repository like DockerHub. With this we have the image fixed with the Fission core services and all the dependencies, ready to create functions that will include the CASA framework.

```
FROM python:3.8-buster
RUN apt-get update -y &&
  apt-get install -y python3-dev libev-dev
WORKDIR /app
COPY requirements.txt /app
RUN pip3 install -r requirements.txt
COPY . /app
ENV PYTHONUNBUFFERED 1
ENTRYPOINT ["python3"]
CMD ["server.py"]
```

Code Listing 2: Dockerfile for a customised environment with python and CASA.

The second approach is an example with wsclean a binary imaging application. We need to use another

[4] https://github.com/manuparra/ska-serverless

[5] Fission environments: https://github.com/fission/environments

template that provides support for this kind of environment. When it comes to creating FaaS functions that call specific applications, it will be necessary that the Dockerfile[6] contains the specific installation of the software in the version we want to use. For this case, we prepared an environment for wsclean 3.3 with support for EveryBeam, Dysco and IDG. Then we build the image and upload it to an image repository such as DockerHub.

Once we have the containers built, the next step is to create an environment within Fission. This environment will allow Fission to know how to manage the logic of the function we are going to implement. To do this, just use the Fission CLI and execute both commands from the code listing 3. The first one enables an environment for the functions with CASA and the second one with wsclean.

```
fission environment create --name \
  python-casa-6.5 --image dockerhub/casa
fission environment create --name \
  wsclean-3.3 --image dockerhub/wsclean
```

Code Listing 3: Adding FaaS enviroments for CASA and wsclean.

With both environments in place, it is now time to create the logic for the functions in the Fission platform. The creation of the function consists of developing a code in the language selected in the environment (Python and native application in our case), which includes the parameterisation and the starting point from where our function will start executing. For the environment with Python and CASA, we are going to design a pipeline operation to allow radio interferometric image reconstruction, using tclean. Code listing 4 shows how to capture the input parameters with request.get_json(), such as the input/output data and the execution parameters, then we define the function to be executed in particular from CASA ct.tclean and finally the function returns where the output data is stored.

```
from flask import request
import os, json
import casatasks as ct
def main():
  param = request.get_json()
  input = "/data/" + param["Input-MS"]
  output = "/data/" + param["Output-MS"]
  ct.tclean(vis = input,
            imagename= output,
            **param)
  return "/data/" + output
```

Code Listing 4: FaaS Fission function for one operation with tclean.

For the design of the function with wsclean, it can be done natively using a bash code to execute the binary command, but it can also be integrated and called from Python code using subprocess. Both options are perfectly compatible. In our case, for convenience in the interfaces, we will use a wrapper from Python to call wsclean as indicated in code listing 5.

```
from flask import request
import os, json
def main():
  param = request.get_json()
  input = "/data/" + param["Input-MS"]
  parameters_str = ... # Extract parameters
  subprocess.run(["wsclean"] +
    parameters_str.split() + [input])
  return "/data/" + output
```

Code Listing 5: FaaS Fission function for one operation with wclean.

For this testbed, we have used an already prepared data that can be downloaded from CASA repository[7]. Finally, one more operation must be carried out, which consists of adding the function to the platform and publishing it in the catalog. In the code listing 6, our functions are integrated and published in two URLs to be able to consume each function independently, in this case in our server *https://server/function/* where function is *tclean or wsclean*.

```
fission fn create --name tclean --env
python-casa \
--code tclean.py --method POST --url "/
tclean/"
```

[6] Customised environment for wsclean: https://github.com/manuparra/ska-serverless/tree/main/Fission/enviroments/wsclean

[7] https://casaguides.nrao.edu/index.php?title=VLA_CASA_Imaging-CASA6.2.0

```
fission fn create --name wsclean --env
 wsclean \
--code wsclean.py --method POST --url
"/wsclean/"
```

Code Listing 6: Two functions created and published in Fission.

To perform some tests, we can call the functions created from the command line by accessing the URL, previously generated and sending it the data and parameters, as indicated in the code listing 7.

```
curl -X POST -d "$(cat parameters-
tclean.json)" \
   -H "Content-Type: application/json" \
   "http://${OURFaaSPlatform}/tclean/"
```

Code Listing 7: Test tclean function with data and parameters.

With all of this procedure, it is possible to expand the function catalog with virtually any type of programming language, application or container so that through the underlying orchestrator, these functions can be scaled indefinitely and interoperated from a workflow or API. The scientific user can publish their functions in source code and containers from a public repository so that they can be shared within the community and then integrated into FaaS.

In terms of metrics, we have tested these functions in a basic way with a small dataset example, obtaining consistent times with a minimal overhead due to management on the orchestrator. These executions can be found in the project repository, and this study is proposed as part of future work where it will include benchmarks to showcase performance in more complex and realistic working environments.

4. CONCLUSIONS AND FUTURE WORK

The SKA, currently under construction, will be the largest radio interferometer, and will be the largest producer of public data on Earth. The SKA Regional Centre will be a federated network of research and computing facilities dedicated to enable researchers to convert SKA data to advanced data products and finally to scientific results. This data processing challenge requires innovative, efficient and robust systems able to scale the workflows while giving enough flexibility to adapt to the complexity of the data products and the necessities of specific research programs. In this context, as an advantage over using containers, FaaS abstracts away infrastructure management and scalability, providing greater portability and flexibility by allowing functions to run on different platforms and environments without additional changes.

We have been aiming to implement a platform for FaaS and create different operations of a radio astronomy workflow as functions available in a catalogue that can be called from virtually any API, service or library, so that we can abstract both the software that runs internally in the function and the scaling of the computational resources it uses. We have encapsulated existing code/libraries and applications within containers to be distributed globally through a deployment of federated orchestrators, with the aim of ensuring the high availability and efficiency of functions executed for example on a large scale as is the SRCNet. This approach is in line with the model of moving computation to where data are located, resulting in reduced latency and improved overall performance.

FaaS still needs to be explored in detail to understand its constraints such as, per-function memory scaling, capabilities similar to AWS SnapStart, and proxy services for low-latency access, databases or data/computing co-location. Therefore, as future work, we propose different aspects to be worked on within the context of SRCNet. On the one hand, it would be interesting to test the performance of the functions developed to monitor performance with data volumes of similar orders of magnitude to those available in SRCNet. Additionally, on the other hand, we propose studying how to implement an »execution planner« that can deliver functions as close as possible to the data, as a key element for SRCNet.

5. ACKNOWLEDGMENT

The authors acknowledge the Spanish Prototype of an SRC (SPSRC) service and support funded by the Spanish Ministry of Science, Innovation and Universities, by the Regional Government of Andalusia, by the European Regional Development Funds and by the European Union NextGenerationEU/PRTR. The SPSRC acknowledges financial support from the State Agency for Research of the Spanish MCIU through the »Center of Excellence

Severo Ochoa« award to the Instituto de Astrofísica de Andalucía (SEV-2017-0709) and from the grant CEX2021-001131-S funded by MCIN/AEI/10.13039/501100011033. Carlos Rios acknowledges Scholarships of introduction to research, »JAE Intro ICU« funded by CSIC, »Programa JAE«.

6. REFERENCES

i T. Lynn, P. Rosati, A. Lejeune, and V. Emeakaroha, »A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms,« *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Hong Kong, China, 2017, pp. 162-169. https://doi.org/10.1109/CloudCom.2017.15

ii S. Sánchez-Expósito, P. Martín, J. E. Ruiz, L. Verdes-Montenegro, J. Garrido, R. Sirvent, A. Ruiz Falcó, R. M. Badia, and D. Lezzi, »Web Services as Building Blocks for Science Gateways in Astrophysics,« *J. Grid Computing*, vol. 14, pp. 673-685, 2016. https://doi.org/10.1007/s10723-016-9382-y

iii H. K. Andi, »Analysis of serverless computing techniques in cloud software framework,« *J. ISMAC*, vol. 3, pp. 221-234, 2021. https://doi.org/10.36548/jismac.2021.3.004

iv A. Barrak, F. Petrillo, and F. Jaafar, »Serverless on Machine Learning: A systematic mapping study,« in *IEEE Access*, vol. 10, pp. 99337-99352, 2022. https://doi.org/10.1109/ACCESS.2022.3206366

v G.C. Fox, V. Ishakian, V. Muthusamy, A. Slominski, »Status of Serverless Computing and Function-as-a-Service (FaaS) in Industry and Research,« 2017, *arXiv:1708.08028*. https://doi.org/10.48550/arXiv.1708.08028

vi M. S. Aslanpour, A. N. Toosi, C. Cicconetti, B. Javadi, P. Svarski, D. Taibi, M. Assuncao, S. S. Gill, R. Gaire, and S. Dustdar, »Serverless Edge Computing: Vision and Challenges,« in *ACSW ‘21: Proceedings of the 2021 Australasian Computer Science Week Multiconference*, Art. no. 10, 2021. https://doi.org/10.1145/3437378.3444367

vii J. Kijak, P. Martyna, M. Pawlik, B. Balis, and M. Malawski, »Challenges for Scheduling Scientific Workflows on Cloud Functions,« in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, 2018, pp. 460-467. https://doi.org/10.1109/CLOUD.2018.00065

viii M. Malawski, A. Gajek, A. Zima, B. Balis, and K. Figiela, »Serverless execution of scientific workflows: Experiments with Hyperflow, AWS Lambda and Google Cloud Functions, *Future Generation Computer Systems*, vol. 110, pp. 502-514, 2020. https://doi.org/10.1016/j.future.2017.10.029

ix Q. Lap Trieu, B. Javadi, J. Basilakis, A. N. Toosi, »Performance Evaluation of Serverless Edge Computing for Machine Learning Applications,« 2022, *arXiv:2210.10331*. https://doi.org/10.48550/arXiv.2210.10331

x R. Kale and C. H. Ishwara-Chandra, »CAPTURE: a continuum imaging pipeline for the uGMRT,« *Experimental Astronomy*, vol. 51, pp. 95-108, 2021. https://doi.org/10.1007/s10686-020-09677-6

xi J. P. McMullin, B. Waters, D. Schiebel, W. Young, and K. Golap, »CASA Architecture and Applications,« in *Astronomical Data Analysis Software and Systems XVI*, in ASP Conference Series, vol. 376, R. A. Shaw, F. Hill, and D. J. Bell, Eds., 2007, pp. 127-130. [Online]. Available https://www.aspbooks.org/publications/376/127.pdf

xii Moldon, Javier, »eMCP: e-MERLIN CASA pipeline,« *Astrophysics Source Code Library*, 2021. ascl:2109.009.

xiii G. I. G. Józsa, S. V. White, T. Kshitij, O. M. Smirnov, P. Serra, M. Ramatsoku, A. J. T. Ramaila, S. J. Perkins, D. C. Molnár, S. Makhathini, F. M. Maccagni, D. Kleiner, P. Kamphuis, B. V. Hugo, W. J. G. de Blok, and L. A. L. Andati, »CARACal: Containerized Automated Radio Astronomy Calibration pipeline«, *Astrophysics Source Code Library*, 2020. ascl:2006.014.

xiv S. van der Tol, B. Veenboer, and A. R. Offringa, »Image Domain Gridding: a fast method for convolutional resampling of visibilities,« *Astronomy & Astrophysics*, vol. 616, Art. no. A27, 2018. https://doi.org/10.1051/0004-6361/201832858

xv A. R. Offringa, J. J. van De Gronde, and J.

B. T. M. Roerdink, »A morphological algorithm for improving radio-frequency interference detection,« *Astronomy & Astrophysics*, vol. 539, Art. no. A95, 2012. https://doi.org/10.1051/0004-6361/201118497

xvi J. Garrido, L. Darriba, S. Sánchez-Expósito, M. Parra-Royón, J. Moldón, M. Á. Mendoza, S. Luna-Valero, A. Alberdi, I. Márquez, and L. Verdes-Montenegro, »Toward a Spanish SKA Regional Centre fully engaged with open science,« *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 8, no. 1, Art. no. 011004, 2022. https://doi.org/10.1117/1.JATIS.8.1.011004

THE GHGA METADATA CATALOG

MAKING HUMAN OMICS DATA FINDABLE

MORITZ HAHN, JENS KRÜGER*, JORDY D. ORELLANA FIGUEROA, ZEHRA HAZAL SEZER, THOMAS JAKOB ZAJAC
High Performance and Cloud Computing Group, Eberhard Karls Universität Tübingen,
Tübingen, Germany, *jens.krueger@uni-tuebingen.de

MANIKANDAN RAVICHANDRAN
GHGA Office, DKFZ German Cancer Research Center Heidelberg, Germany

BILGE SURUN
Applied Bioinformatics, Institute for Bioinformatics and Medical Informatics,
Eberhard Karls Universität Tübingen, Tübingen, Germany

ANANDHI IYAPPAN
Structural and Computational Biology Unit, EMBL, Heidelberg, Germany

KAROLINE MAUER
Systems Medicine (DZNE), Precise Platform for Genomics and Epigenomics
(DZNE, Universität Bonn), DZNE, Universität Bonn, Bonn, Germany

SVEN NAHNSEN
Quantitative Biology Center, Eberhard Karls Universität Tübingen, Tübingen, Germany

GALINA TREMPER
Federated Information Systems (DKFZ), Complex Data Processing in Medical Informatics (UMM), German Cancer Research Center (DKFZ), University Medical Center Mannheim (UMM), Heidelberg/Mannheim, Germany

ABSTRACT

Advancements in the collection of omics data have led to the generation of large datasets of human omics data for health research, resulting in the emergence of various data sharing platforms to enable researchers to access broader collections of data. However, there currently is a lack of a suitable platform in Germany. To address this, the German Human Genome-Phenome Archive (GHGA) project aims to contribute to research possibilities by developing an archive for human omics data using FAIR principles as part of the Federated European Genome-Phenome Archive (FEGA). The GHGA project intends to facilitate data sharing across international borders while strictly protecting individuals' sensitive data and privacy. In this paper, we describe the GHGA Metadata Catalog, previously launched as GHGAs first milestone, serving as a discovery platform for human omics data. While not directly providing data yet, it does heighten data visibility and connects Researchers with Data Providers.

Keywords: *Human Genome Data; Omics; Science Gateway; FAIR; Sensitive Data; NFDI; FEGA*

1. INTRODUCTION

The German Human Genome-Phenome Archive (GHGA)[1] is part of the German National Research Data Infrastructure (NFDI)[2,i] and is committed to supporting researchers in biomedical research who work with human omics data. The NFDI initiative aims to establish systematic data management for research data in all academic disciplines by providing long-term data storage, backup, accessibility, and network of the data across borders, adhering to FAIR principles (findable, accessible, interoperable, reusable data)[ii] for all individual NFDI consortia. GHGA is dedicated to generating additional value for their individual communities through well-structured data management. The collection of omics data from participants is critical for biomedical research, with

[1] https://www.ghga.de

[2] https://www.nfdi.de/

cc BY NC ND
https://dx.doi.org/10.15496/publikation-100323

many applications in biology, translational research, and medicine. High-volume data generation for personalized therapies is increasing daily, expanding the toolset for precision diagnostics. The fast-growing amount of data presents both an opportunity for research as well as challenges for handling the infrastructure needs associated with the data. GHGA aims to build a national infrastructure for the storage and processing of human omics data in a uniform, data protection-compliant framework. GHGA is also a national node of the Federated European Genome-Phenome Archive (FEGA)[iii], allowing for human omics data to be accessible and optimally usable for national and international research while adhering to national regulations on data protection. GHGA also plans to establish an efficient, easy-to-use, large-scale analysis infrastructure for biomedical research to address the practical needs of the research community.

The anticipated functionality of GHGA is divided into high-level milestones, and each active milestone is broken down into smaller work packages and tasks utilizing an agile methodology. GHGA has four primary milestones, namely Metadata Catalog, Archive, Cloud and Atlas. Recently, the GHGA Metadata Catalog[3] was released and serves as a discovery platform for human omics data that is available for research purposes. During this early phase of the project, it allows users to browse, search, and filter metadata submitted to GHGA. The GHGA Metadata Catalog is the initial phase towards our objective of offering complete data archival services for human omics data. The datasets within the catalog are annotated using the GHGA Metadata Model[iv], which is compatible with the EGA metadata model. Subsequently, during the GHGA Archive phase, GHGA will establish EGA-like functionality in Germany, structured in a federated manner. This functionality will be broadened to include data processing capability and provide insights into portal content through statistics and aggregations during the GHGA Cloud phase. Lastly, GHGA aims to become a Platform as a Service (PaaS) by providing researchers with tools to access GHGA data and hardware resources for their customized data processing pipelines in the GHGA Atlas phase.

2. TECHNOLOGY

The GHGA Metadata Catalog is based on the technologies of the data portal currently being developed for the GHGA Archive milestone, and thus inherits its technology stack.

The front-end is based on the React 18 JavaScript library[v], with the use of TypeScript language features through an additional library for React[vi]. The look of the GHGA Metadata Catalog is based on the Bootstrap 5.2 library[vii] along with SASS for further customizations, as well as responsive design at all screen sizes.

The GHGA Metadata Catalog runs on the de.NBI Cloud[viii].

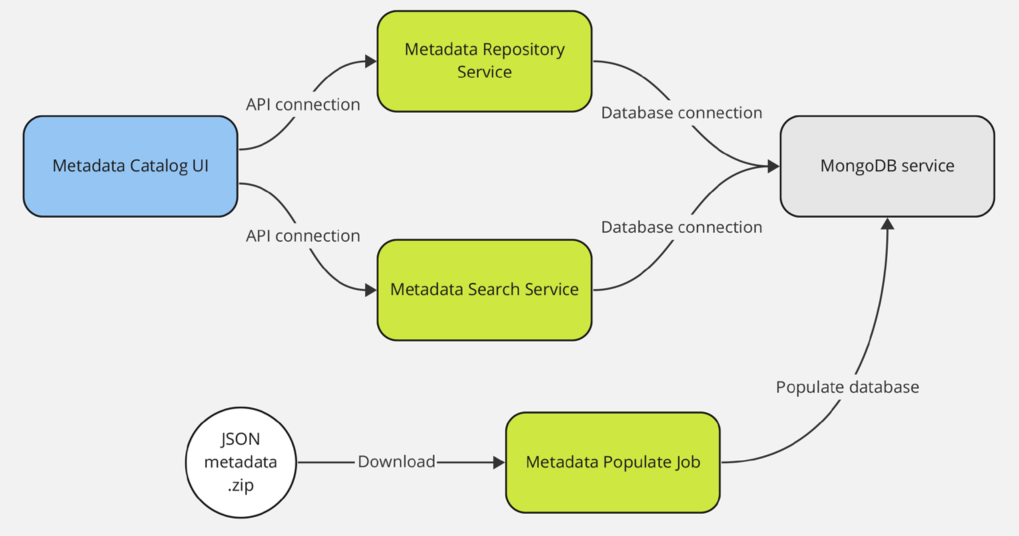


Figure 1: Components and services of the GHGA Metadata Catalog.

[3] https://catalog.ghga-dev.de/

2.A. MICROSERVICE ARCHITECTURE

The GHGA Metadata Catalog uses a microservice architecture consisting of the UI, Metadata Search, Metadata Repository, Metadata Populate Job, and MongoDB microservices (see Figure 1), all of which are open source and hosted on the organization's GitHub[4]. The UI microservice runs the React-based website on a Node.js 16 environment[ix], itself running on Linux Alpine 3.15[x]. The metadata displayed by the Catalog are first obtained from a zip file of JSON files, which are compliant with our metadata model (see section III.C.). The Metadata Populate Job microservice takes these JSON files and populates a database in the MongoDB 5 back-end service[xi]. Both the Metadata Search Service and the Metadata Repository Service connect to the MongoDB service and provide an API for the UI to obtain a list of search results, as well as detailed information on individual metadata documents (as well as summary statistics, such as the total number of datasets) respectively. Further details on the functionality of the Metadata Search and Repository services will follow in the next section.

2.B. THE LINKED DATA MODELLING LANGUAGE (LINKML)

LinkML[xii] is a data modeling framework that uses object-oriented principles (e.g., inheritance) to define data structures. It offers the flexibility to describe domain-specific data models in YAML[xiii] format and enables cross-framework data-model interoperability through artifact generation from YAML to standardized data formats such as JSON[xiv] and RDF[xv].

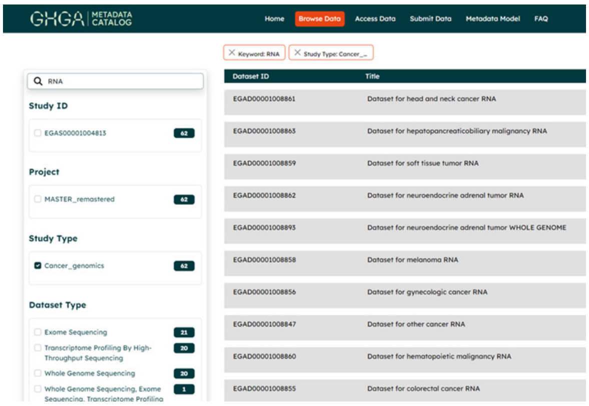
In GHGA we modeled the metadata starting from the »Individual« who is subject to the »Experiment« to the (raw and processed) genomics data together with the data access controller metrics, using a LinkML model which is employed in the metadata backend services. Additionally, we generated extensive documentation and schema visualization, obtained with LinkML.

3. FUNCTIONALITY OF THE GHGA METADATA CATALOG

The metadata currently accessible for browsing are from the clinical study *Comprehensive Genomic and Transcriptomic Analysis of Rare Cancers for Guiding of Therapy (H021)*, available in the European Genome Archive (EGA)[xvi].

3.A. SEARCH FUNCTIONALITY

The Metadata Search Service provides an API to perform database searches through its core functionality of building MongoDB queries. The microservice allows the search of different types of entities (e.g. datasets, studies, samples, experiments, etc.) using either a text-based search, or using a dictionary of facets to filter by (e.g. dataset type: Exome Sequencing). With these last two search options, users of the GHGA Metadata Catalog UI (see Figure 2) can perform text-based searches as well as filter the list of datasets or list of search results using the side panel in the Browse Data page.



RNA

Study ID

□ EGAS00001004813 62

Project

□ MASTER_remastered 62

Study Type

☑ Cancer_genomics 62

Dataset Type

□ Exome Sequencing 21
□ Transcriptome Profiling By High-Throughput Sequencing 20
□ Whole Genome Sequencing 20
□ Whole Genome Sequencing, Exome Sequencing, Transcriptome Profiling 1

Dataset ID	Title
EGAD00001008861	Dataset for head and neck cancer RNA
EGAD00001008863	Dataset for hepatopancreaticobiliary malignancy RNA
EGAD00001008859	Dataset for soft tissue tumor RNA
EGAD00001008862	Dataset for neuroendocrine adrenal tumor RNA
EGAD00001008893	Dataset for neuroendocrine adrenal tumor WHOLE GENOME
EGAD00001008858	Dataset for melanoma RNA
EGAD00001008856	Dataset for gynecologic cancer RNA
EGAD00001008847	Dataset for other cancer RNA
EGAD00001008860	Dataset for hematopoietic malignancy RNA
EGAD00001008855	Dataset for colorectal cancer RNA

Figure 2: Screenshot of the GHGA Metadata Catalog webpage, depicting content and functionality.

The text search not only looks through dataset titles, but every field of every metadata item in the database and obtains the specific document(s) that match the search query. For example, a text search for a specific file name, an experiment ID, sample phenotype, or part of a study description will all succeed and show the user all applicable datasets that contain the file name, experiment, sample, or study in question.

At the moment, the string-based text-search is not yet very powerful, resulting in a couple of limitations. For example, searching for *Dataset* will provide

[4] https://github.com/ghga-de

results, searching for *Datas* or *Dat* will not. Also, a query such as *neck cancer RNA* will provide one result in the *Dataset for head and neck cancer RNA*, but the query *neck RNA* or *RNA cancer* will provide no results, as the search strings cannot be found for any dataset, even if the words that compose the search string are evidently present.

These limitations will be addressed in the GHGA Archive data portal through the implementation of a more robust search engine.

3.B. METADATA REPOSITORY SERVICE

As mentioned previously, the Metadata Repository Service provides an API to obtain the information for a single specific metadata document. It also allows for the creation of several types of metadata documents, such as datasets and submissions, though this functionality is not used by the Catalog. The service can also provide summary statistics for the entire metadata in the database, such as the total number of studies, all the file types and the total files for each file type available, as well as the total number of individuals by sex. The summary statistics are also available for each individual dataset, providing similar information as before on a per-dataset basis.

3.C. METADATA MODEL

The GHGA Metadata Model is a central aspect to the harmonization and standardization efforts undertaken by GHGA. It serves the discovery and (re-) use of data while following the General Data Protection Regulation (GDPR)[xvii]. For this, GHGA has developed a consent toolkit which includes a data sharing module. This is not limited to GHGA and describes a way of sharing data securely for scientific research[xviii].

By exploring and building on several already existing models and in close discussions with stakeholders from genomics medicine, we defined a harmonized metadata model covering metadata elements pertaining to »Technical«, »Individual«, and »Dataset« data. Standardization of the model is achieved with the usage of several well-established ontologies and the definition of controlled vocabularies, making it self-describing, unambiguous, flexible, and expressive as well as complying with the FAIR principles. The ontologies were chosen based on their suitability to represent the knowledge specific to genomic medicine. They have wide acceptance and community support, which increases their interoperability and reusability. The backbone of the schematic model is defined in YAML using LinkML and built incrementally to ensure that (i) the schema has a basic core that

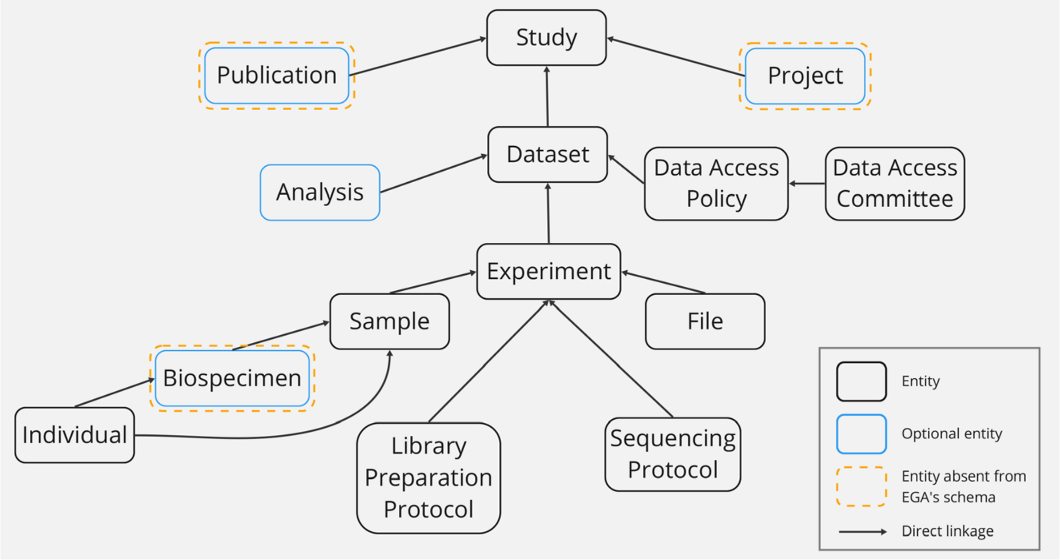


Figure 3: Overall representation of GHGA Metadata Schema.

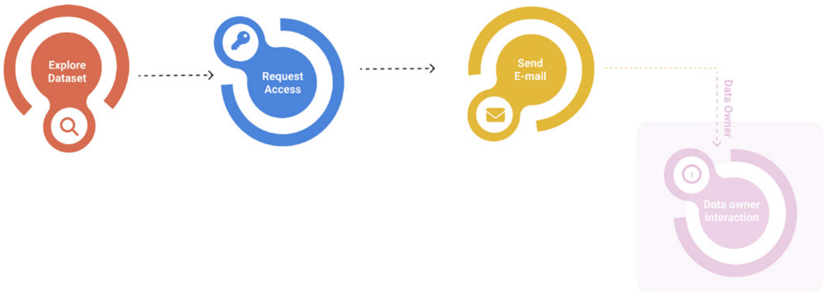


Figure 4: High level depiction of the workflow for accessing data.

is robust and (ii) additional extensions to the schema can be easily adapted and extended according to the requirement of future use cases.

One of the core objectives of the metadata model is to provide a trustworthy long-term archive of human omics data federated with EGA. Therefore, the core structure of the GHGA metadata schema is closely aligned to the EGA metadata schema[xix]. GHGA's prototype model was developed in close alignment to the EGA model. However, as data protection regulations differ between Germany and England/Spain, who host central EGA, GHGA's model must be adapted to fulfill the respective GDPR requirements. Currently, EGA stores both *Sample-* and *Individual*-related metadata in their *Sample* entity. To be GDPR-compliant, GHGA introduced the *Individual* and *Sample* entities, which distinguish between sensitive, personally identifiable information and public metadata, respectively.

The overall structure and metadata type acquisition is closely aligned between the two models. Entities are the building blocks of the GHGA metadata schema. Entities represent real world objects, such as *Study* or *Sample*, that capture certain aspects of metadata. Each entity is represented by several properties that provide further detail about it (*e.g.* the properties *age* and *sex* describe the *Individual* entity). The GHGA schema captures the following entities, which are also present in the EGA schema: *Study, Dataset, Analysis, Data Access Policy, Data Access Committee, Experiment, Library Preparation Protocol, Sequencing Protocol, File* and *Sample*. GHGA further introduced three entities which are not present in the EGA schema: *Project, Publication* and *Biospecimen* as seen in Figure 3. All three entities are considered *Optional* during submission since not every submission might already be linked to a *Publication* or a *Project* at the time of submission, nor will every submission contain a *Biospecimen*. If available, metadata can be added at a later point in time. However, the above three entities are part of the core schema because they increase the FAIRness of a dataset.

EGA's schema is highly interlinked to create multiple access points for EGA's metadata API. The linkage in GHGA's metadata model is reduced and creates a linear, hierarchical model with the *Study* as the central entity. This ensures a guided flow of information from one entity to another and is in accordance with

The GHGA and EGA schema share the same ontologies which smoothens data transfers and makes data exchange interoperable. Both schemas allow widely accepted ontologies such as Human Phenotype Ontology (HPO)[xx] or National Cancer Institute Thesaurus (NCIt)[xxi], as well as the Global Alliance for Genomics and Health (GA4GH) standard Data Use Ontology (DUO)[xxii]. For properties where ontologies are currently unavailable, we defined lists of controlled vocabularies which are also closely aligned to vocabularies allowed by EGA. Approximately 70% of the mandatory properties captured in the GHGA model are controlled with either an ontology term, controlled vocabularies, or a data type definition. The remaining 30% capture information that is currently difficult to standardize, such as the Data Access Policy Text which is mostly free text. Further measures need to be taken to standardize these properties.

3.D. ACCESSING DATA

Users can determine if a dataset meets their requirements by using the browser and filter functionalities within the GHGA Metadata Catalog. They can then proceed to the relevant dataset and select the »Request Access« option. Doing so triggers the users'

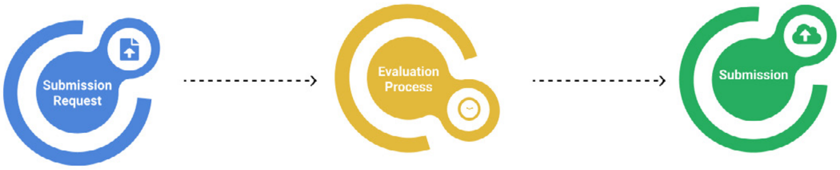


Figure 5: High level depiction of the workflow for submitting data.

email client to open and display an email template that is directed to the dataset's Data Access Committee or responsible individual. The users furnish the requisite information in the email and send it to the email address listed for the data access committee. Any data exchange in the GHGA Metadata Catalog then happens directly between the Data Requester and the Data Owner. It is also important to bear in mind that GHGA refrains from engaging in the data access negotiation process (see Figure 4).

3.E. SUBMITTING DATA

It is well known that establishing common practice in genomic data sharing is a long and strenuous process. In order to simplify the data submission process, we have designed a two-dimensional representation of our model as a submission spreadsheet which captures all the metadata fields that are part of the metadata model. The submission spreadsheet aims to be user-friendly and allows the data submitters to provide their data without prior knowledge in information technology or computer science.

After receiving the filled spreadsheet from the data controller, the next steps of submission are performed on the side of the GHGA Metadata Catalog. The metadata in the spreadsheet will be validated against the JSON representation. The resulting JSON file can be submitted to the Metadata Repository Service (MRS) which registers and interlinks the entities within the submission and stores them to the MongoDB Database within the dedicated collections. Finally, after successful submission to the MRS, the data will be displayed in the Catalog. Data stewards will assist with the entire process of data submission and check the provided metadata for structural integrity, completion and that the content doesn't lead to the re-identification of patients. The last thing is contractually assured by the data owner, so that they do not give us personal metadata, but we will likely still double check.

Data owners who prefer to provide the metadata in JSON format can create the input JSON file by themselves and skip the transpilation step (transpiler services involve the process of parsing a code in one format and generate an equivalent version in another format). The provided JSON file should correspond exactly to our metadata model and include all the required files. The received JSON will be validated internally to assure the correctness, and the submitter will be given feedback if the format is invalid. For this submission workflow the submitter needs to have a deep understanding of the model. The valid JSON files are submitted to the metadata repository in the same way as for the complete submission workflow. The stored metadata can be later accessed and queried using the Metadata Search Service. The query results can be also seen on the Catalog webpage.

4. CONCLUSION

In this paper, we described the GHGA Metadata Catalog, which serves as a discovery platform for human omics data. As GHGA's first milestone, it represents a significant step forward in achieving GHGA's goal of providing FAIR access to human omics data. It provides heightened visibility by providing search functionalities over already available datasets and connecting data providers with researchers.

Looking towards the future, the GHGA Archive, currently under development and set to release in the second half of 2023, will add full FEGA capabilities by storing research data itself and therefore making data sharing even easier—while data providers still retain full control over their shared dataset. GHGA Archive will also address possible limitations, such as the current search function experienced while working with the GHGA Metadata Catalog. Fulfilling the NFDI's mission in the field of human omics research, GHGA already has a large potential for advancing genomic research in Germany and beyond. The GHGA Metadata Catalog serves as a crucial resource for researchers seeking to discover available data sets and connect with data providers, and the GHGA Archive will facilitate even easier data access.

5. ACKNOWLEDGMENT

We represent only a small fraction of the GHGA consortium. We thank all colleagues and friends from GHGA for their continuous support. We acknowledge support for GHGA from the German National Research Data Initiative (441914366). We also acknowledge the use of the de.NBI Cloud and the support by the Federal Ministry of Education and Research (BMBF) through grant no 031A535A.

6. REFERENCES

i N. Hartl, E. Wössner, and Y. Sure-Vetter, »Nationale Forschungsdateninfrastruktur (NFDI),« *Informatik Spektrum*, vol. 44, no. 5, pp. 370–373, 2021. https://doi.org/10.1007/s00287-021-01392-6

ii M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J-W. Boiten, L. Bonino da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Corsas, I. Dillo, O. Dumon, S. Edmonds, C. T. Evolo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Growth, C. Goble, J.S. Grethe, … and B. Mons, »The FAIR Guiding Principles for scientific data management and stewardship,« *Scientific Data*, vol. 3, Art. no. 160018, 2016. https://doi.org/10.1038/sdata.2016.18

iii I. Lappalainen, J. Almeida-King, V. Kumanduri, A. Senf, … A. Navaro, and P Flicek, »The European Genome-Phenome Archive of human data consented for biomedical research,« *Nature Genetics*, vol. 477, pp. 692–695, 2015. https://doi.org/10.1038/ng.3312

iv The GHGA Team, Sept. 22, 2022, »GHGA Metadata Schema, version 0.9.0.« [Online]. Available: https://github.com/ghga-de/ghga-metadata-schema

v The React Team, Mar. 29, 2022, »React, version 18.0.« [Online]. Available: https://react.dev

vi DefinitelyTyped, Nov. 4, 2022, »DefinitelyTyped, version 18.0.25.« [Online]. Available: https://github.com/DefinitelyTyped/DefinitelyTyped

vii Bootstrap Authors [https://github.com/twbs/bootstrap/graphs/contributors], Jul. 19, 2022, »Bootstrap, version 5.2.« [Online]. Available: https://getbootstrap.com

viii P. Belmann, B. Fischer, J. Krüger, M. Procházka, H. Rasche, M. Prinz, M. Hanussek, M. Lang, F. Bartusch, B. Gläßle, J. Krüger, A. Pühler A, and A. Sczyrba, »de. NBI Cloud federation through ELIXIR AAI,« *F1000Research*, vol. 8, Art. no. 842, 2019. https://doi.org/10.12688/f1000research.19013.1

ix The Node.js Docker Team, Sept 8, 2023, »The official Node.js docker image, version 16-bullseye.« [Online]. Available: https://hub.docker.com/_/node?tab=description&name=16-bullseye

x Alpine Linux Development Team, Nov. 24, 2021, »Alpine Linux, version 3.15.0.« [Online]. Available: https://alpinelinux.org

xi MongoDB, Inc., July 13, 2021, »MongoDB, version 5.0.« [Online]. Available: https://www.mongodb.com

xii S. Moxon, H. Solbrig, D. Unni, D. Jiao, R. Bruskiewich, J. Balhoff, G. Vaidya, W. Duncan, H. Hedge, M. Miller, M. Brush, N. Harris, M. Haendel, and C. Mungall, »The Linked Data Modeling Language (LinkML): A General-Purpose Data Modeling Framework Grounded in Machine-Readable Semantics,« *CEUR Workshop Proceedings,* vol. 2073, pp. 148–151, 2021.

xiii O. Ben-Kiki, C. Evans, C., and B. Ingerson, B., 2009, »Yaml ain't markup language (yaml™), version 1.1, Working Draft 2004-12–28.« [Online]. Available: https://yaml.org/spec/history/2004-12-28/2004-12-28.pdf

xiv T. Bray, Ed., »The javascript object notation (json) data interchange format,« Request for Comments 7159, Mar. 2014. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7159

xv D. Brickley and R. V. Guha, Eds., »RDF Schema[, version] 1.1,«. W3C recommendation, Feb. 25, 2014. [Online]. Available: https://www.w3.org/TR/rdf-schema/

xvi P. Horak, C. Heining, S. Kreutzfeldt, B. Hut-

ter, A. Mock, J. Hüllein, M. Fröhlich, S. Uhrig, … H. Glimm, and Stefan Fröhling, »Comprehensive Genomic and Transcriptomic Analysis for Guiding Therapeutic Decisions in Patients with Rare Cancers,« *Cancer Discovery*, vol. 11, no. 11, pp. 2780–2795, 2021. https://doi.org/10.1158/2159-8290.CD-21-0126

xvii European Commission. »General Data Protection Regulation (GDPR) Compliance Guidelines.« [Online]. Available: https://gdpr.eu/

xviii A. Bruns, A. Benet-Pages, J. Eufinger, H. Graessner, O. Kohlbacher, F. Molnár-Gábor, S. Parker, C. Schickhardt, O. Stegle, and E. Winkler, »Consent Modules for Data Sharing via the German Human Genome-Phenome Archive (GHGA), version 1.0,« Zenodo, July 13, 2022. https://doi.org/10.5281/zenodo.6828131

xix EGA, Apr. 26, 2021, »EGA Metadata Schema, version 1.0.0.« April 26, 2021. [Online]. Available: https://github.com/EbiEga/ega-metadata-schema

xx P. N. Robinson, S. Köhler, S. Bauer, D. Seelow, D. Horn, and S. Mundlos, »The Human Phenotype Ontology: a tool for annotating and analyzing human hereditary disease,« *The American Journal of Human Genetics*, vol. 83, no. 5, pp. 610–615, 2008. https://doi.org/10.1016/j.ajhg.2008.09.017

xxi N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W-L. Shaiu, and L. W. Wright, »NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information,« *Journal of Biomedical Informatics*, vol. 40, no. 1, pp. 30–43, 2007. https://doi.org/10.1016/j.jbi.2006.02.013

xxii J. Lawson, M.N. Cabili, G. Kerry, F. Boughtwood, A. Thorogood, … and M. Courtot, »The Data Use Ontology to streamline responsible access to human biomedical datasets,« *Cell Genomics*, vol. 1, no. 2, Art no. 100028, 2021. https://doi.org/10.1016/j.xgen.2021.100028

INCENTIVISED RECOMMENDATION FRAMEWORK OF INTERNET OF THINGS DATA PROVISION FOR SCIENTIFIC RESEARCH

SANDI GEC*, VLADO STANKOVSKI, DEJAN LAVBIČ, PETAR KOCHOVSKI
Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
*sandi.gec@fri.uni-lj.si

ANDREJ KOS, URBAN SEDLAR
Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia

ABSTRACT

Data collection in the domain of Internet of Things (IoT) devices, such as smart wearable devices, utility devices containing sensors and other sensor-based devices used in everyday tasks, is often centralised or decentralised among specific interest groups. In the scientific field, the availability of IoT data may often be challenging due to the lack of incentive from data owners (e.g. users of wearable IoT devices) to provide data for science gateway purposes. In this paper, we propose a recommendation framework for the incentivised distribution of IoT data developed on top of ShimmerEVM, a distributed ledger using directed acyclic graph of blocks called Tangle and at the same time supports Ethereum Virtual Machine (EVM) smart contracts. We propose a two-step approach enabled by dedicated smart contracts running on top of IOTA Tangle EVM enabling ledger to comprehensively facilitate the IoT data provision to the research groups as a science gateways tool. In the first step, we propose a data management recommendation algorithm to select the relevant datasets based on the requirements. In the second step we developed an IoT data provision incentivisation smart contract communicating among data providers (e.g. users of wearable smart devices) with tokenised rewards and research groups using the data for scientific purposes. The results indicate the feasibility and sustainability of our approach.

Keywords: *Internet of Things; Incentive; Smart Contract; Ethereum Virtual Machine; Recommendation Framework*

1. INTRODUCTION

The Internet of Things (IoT) has become a ubiquitous part of our daily lives, generating massive amounts of data. This data has the potential to revolutionise scientific research and enable breakthroughs in various fields[i]. However, the data's sensitive nature and the challenges associated with incentivising users to provide it for research purposes remain major barriers. Decentralised ledger technology (DLT), particularly blockchain, has been identified as a promising solution to these challenges. While blockchain technology has matured from Cloud-to-Edge and its portability of components, different types of blockchain still present challenges for privacy and security. Additionally, introducing trustless mechanisms for incentivisation has proven to be difficult. Thus, the provision of the potential research data to the research groups is very important in the scope of science gateways.

In this paper, we propose an incentivised recommendation framework for science gateways data resources using IOTA Tangle DLT and smart contracts to overcome these limitations. From the perspective of science gateways, we strive to establish a tight collaboration among users of wearable devices (data providers) and research groups performing research on the obtained data. Our approach utilises a smart contract incentivisation mechanism to encourage fundamental interactions among users within the system. We believe that our framework has the potential to address the privacy and security challenges associated with IoT data provision for scientific research while providing a trustworthy and incentivised data sharing platform.

The remainder of this paper is structured as fol-

https://dx.doi.org/10.15496/publikation-100323

lows. Section 2 places our work in the context of the state-of-the-art. Section 3 describes the main use case scenario. Section 4 outlines the key features of the recommendation framework from the perspective of potential DLTs background analysis, the high-level architecture, smart contract-based recommendation algorithm and implementation details. Section 5 presents the experimental evaluation study, and Section 6 draws discussion with conclusions.

2. RELATED WORK

IoT domain, independent of the subdomain, has a general purpose which can be summarised as record of sensor data used for specific purposes. Data owners are not a priori incentivised to share such data due to privacy and security concerns. Therefore Maddikunta et al[ii] proposed an overview of recently available incentive techniques to improve the domains and/or encourage data provision on IoT subdomains using different techniques where smart contract techniques are suitable for privacy improvement (e.g. federated learning, vehicles). Before the introduction of smart contracts in the IOTA ledger[iii] an access control framework founded on Ciphertext-policy Attribute-Based Encryption was proposed[iv] where the data owners are in charge of authorization. Nevertheless, the proposed solution provides limited scalability. In the announced major update of IOTA 2.0 many improvements are planned to be implemented such as EVM support that is currently available in the testnet environment from March 2023[v]. A smart contract-based access control for IoT data was proposed by Y. Zhang et al.[vi] evaluated on Ethereum ledger. In our work, we propose a recommendation framework that leverages the IOTA EVM-enabling methodology of asset definition that allows the main ledger coin SMR in ShimmerEVM to be used as a token in the smart contract thus allowing more advanced contract functionalities.

In our previous work we analysed the smart contract template library OpenZeppelin[1] and found a detailed support of ready-to-use smart contract functionalities and concepts[vii]. In the review work of blockchain and smart contracts in the IoT domain the authors identified promising research work on blockchain platforms such as Ethereum, Hypeledger, Composer and IOTA Tangle[viii]. Apart from access control and secure data management M. Mihaylov et al.[ix] proposed a blockchain-based reward mechanism in the renewable energy domain.

In this work, we propose a recommendation framework for research data provision among data providers as users of wearable devices and research groups. The framework is built on EVM-enabling IOTA Tangle ledger ShimmerEVM where the recommendations are provisioned through dedicated smart contracts thus making it possible to also support incentivisation of provisioning with tokenisation rewards for data providers.

3. USE CASE SCENARIO

The blockchain technology has introduced revolutionary changes in the context of trustworthiness, traceability, security and privacy of data. Namely, the integration of blockchain technology with the IoT has the potential to bring significant benefits in various domains, such as: increasing data security, providing data integrity proofs, help automate transactions between IoT devices, reduce the need for intermediaries and similar. In this paper we propose a recommendation framework to provision IoT data for scientific research and will establish trustworthy and privacy preserving relationships between the data providers (e.g. users of wearable sensors) and researchers (e.g. healthcare data analysts). In the investigated case, a group of researchers that are studying the impact of physical activity requires access to the patients' data that is accumulated by their wearable smart devices (e.g. smart watches, blood pressure devices, heart rate sensors, etc.). Both user groups are identified and authenticated on the system through blockchain. Moreover, data's integrity and authenticity are also verified on the blockchain. In the following we will describe in more details the scenarios for each user type.

Data providers (i.e. chronic disease patients) use wearable devices to constantly monitor their vital signs by using wearable IoT devices that are assigned unique digital identities. To ensure data's integrity and preserve privacy, the data is encrypted, watermarked and timestamped before it is recorded on the blockchain. Once the data is recorded on the block-

[1] https://www.openzeppelin.com/

chain, it is immutable, hence it cannot be altered by anyone using it. Moreover, by having the data watermarked and encrypted, it can be always verified on its originality and source. By using a user-friendly web interface, the data providers can easily manage access to their sensors' data and also receive incentives in the form of crypto tokens.

Data consumers (e.g. researchers) can use the user interface to request access to specific datasets or subsets of data for their research purposes. Initially they input their requirements (e.g. type of data, amount of data, approximate duration of the access to the data and similar). Based on the given requirements, the recommendation framework retrieves the optimal data of the providers and recommends them to the data consumers. Upon their confirmation, the recommendation framework invokes a smart contract, which is deployed on the blockchain and facilitates peer-to-peer interaction between the data providers and data consumers in the context of temporary data access management. In summary, the smart contract will have the functionality to verify important criteria, such as: (1) whether the data provider has given consent for his/her data to be accessed by data consumers, (2) if the data is authentic, valid and consistent, (3) if the data consumers have the sufficient amount of tokens to perform the transactions on the blockchain, (4) if the data consumer is accessing the data in the predefined time period.

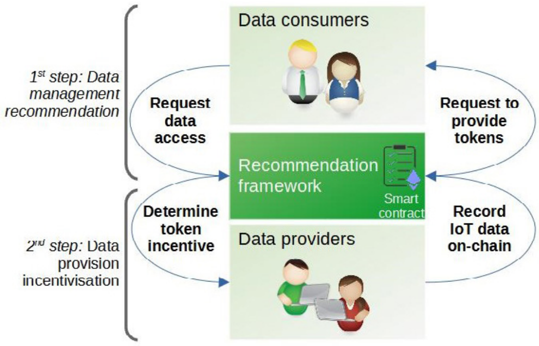

1st step: Data management recommendation
Request data access
Data consumers
Request to provide tokens
Recommendation framework
Smart contract
Determine token incentive
Data providers
Record IoT data on-chain
2nd step: Data provision incentivisation

Figure 1: Overview of the use case scenario.

The technical details on the system that will support the given scenario are described in the following sections and the fundamental workflow is depicted in Figure 1.

4. A RECOMMENDATION FRAMEWORK FOR DISTRIBUTION OF IOT DATA LEVERAGING DECENTRALISED TECHNOLOGIES

This section presents the pillar components of our recommendation framework. First, we present a technical overview of the potential decentralised technologies and their key features. Second, we outline a high-level architecture of our framework. Third, we propose a fully on-chain recommendation algorithm built upon smart contract (1st step in Figure 1) and last, we indicate the implementation details of our system.

4.A. BACKGROUND OF POTENTIAL DECENTRALISED TECHNOLOGIES

The decentralised components such as blockchain ledgers and related technologies allows existing Cloud-to-Edge systems to integrate important features through the ledger typologies and available features. Therefore, we analysed the most prominent ledger technologies suitable for the IoT domain. Since the domain in general generates a big amount of data from sensor devices it is crucial in the first place to allow high throughput. Moreover, to efficiently design a transparent recommendation framework that allows incentivisation of IoT data the ledger should support smart contracts and thus allow the key func-

Distributed Ledger Technology	Type	Throughput	Feature
Ethereum	Blockchain	Low	Smart contracts, high fees
EVM-Enabling Ledgers	Blockchain	Medium	Smart contracts, low fees
Hyperledger Besu	Key-Value Store	High	Private network, smart contracts, apis, monitoring, feeless or minimal fees
IOTA	Tangle	High	Feeless, scalable
ShimmerEVM	Tangle	High	Smart contracts, scalable, call views api, minimal fees

Table 1: Background of the most prominent DLTs for IoT domain.

tionalities to be fully transparent and consequently operational on-chain. All the emphasised key requirements are available in the Distributed Ledger Technologies (DLTs) as depicted in Table 1.

In our work, we selected ShimmerEVM that was recently introduced in the 1st quarter of 2023, currently available as a testnet environment. The selected DLT is public and based on Tangle ledger type where the throughput is high because Tangle enables parallel validation of transactions without requiring total ordering. ShimmerEVM also enables EVM-based smart contract where the extended EVM functionalities are summarised as follows:

- Native coin, the asset SMR, is conveniently supported as a token (e.g. ERC-20, ERC-721) within the smart contract.
- There are many interfacing contracts that allow you to use native assets deposited to a chain within EVM without losing the flexibility to transfer those assets back to Level-1 if desired.

4.B. HIGH-LEVEL ARCHITECTURE

Recommendation framework is an environment built upon four main components: (i) front-end, (ii) back-end, (iii) IOTA Tangle (ShimmerEVM) and (iv) IoT wearable devices. External entities interacting with the framework components are data providers and data consumers as described in the previous section. The high-level architectural overview is depicted in Figure 2.

Front-end is Web based and allows data consumers user-friendly interaction among other components. For example, the user does not need to know technical aspects of smart contracts and/or other Web 3.0 functionalities (e.g. events) since the Web Graphical User Interface (GUI) facilitates all DLTs interactions. The main idea of the front-end is to establish an agreement for the potential token reward that data consumers are willing to pay in order to receive the IoT data from the data providers. To minimise the overhead data a recommendation algorithm is proposed where the data consumers list independent requirements. Therefore, data consumers are served with a subset of IoT data that varies in the sampling frequency and the category of the IoT data.

Back-end interprets the on-chain in the database data of the system users, monitors on-chain transactions, by mainly listening on smart contract events, storing the smart oracle metadata and in general stores the relevant off-chain data in a database.

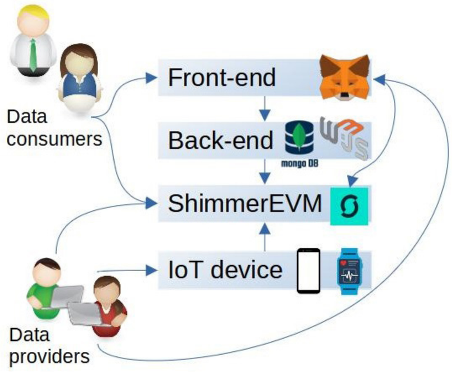


Figure 2: High-level architecture displaying the main system components and their user types.

The system DLT EVM-enabling ShimmerEVM provides a transparent, secure and programmable IoT data provision by leveraging dedicated smart contracts. This component is crucial in the IoT data provision since the IoT devices store the IoT data in the DLT. In case of scalability and performance concerns it is suitable to write IoT data in batch with less transactions but more data or store on-chain only hashed results of the data where the actual IoT data may be stored off-chain on the premises of the data provider on the device or in a centralised or a distributed database. The purpose of the hashes is to guarantee the integrity of the IoT data. Another important functionality is the smart contract for incentivisation, presented in the following Section, that ensures the execution of the agreement.

In the previous section the use case scenario described the IoT subdomain of wearable devices such as dedicated medical devices monitoring vital signs and common wearable devices commonly known as smart devices (e.g. mobile phones, tablets, smart watches etc.). Both types of devices need to be either (1) programmable to allow deploying custom applications (e.g. Android, iOS) and/or (2) allowing common IoT protocol support (e.g. REST API, AMQP, MQTT and others). In the case of applications, it is mandatory for data providers to install the monitoring application and determine the provisioning policy within a smart contract. In case when deploying applications in the jurisdiction of the data providers is not possible the data provision has to be established through a Gateway component. This paper focuses on the (1st) case of deploying an Android application that is further described in the implementation details subsection and in the next section.

4.C. IMPLEMENTATION DETAILS

We implemented our proposed framework using a combination of Sensify[2], Angular[3], MongoDB[4], OpenZeppelin[5], and web3.js[6]. Sensify is an open-source Android application that provides an end-to-end solution for managing IoT data in mobile and wearable devices. Angular is a widely used front-end framework that provides a user-friendly interface for interacting with the platform. MongoDB is a popular NoSQL database that is used to store the data collected from IoT devices.

We also utilised OpenZeppelin, a popular smart contract library, to develop the smart contracts used in our incentivisation mechanism. OpenZeppelin provides a collection of reusable smart contracts that are thoroughly tested and audited. Finally, we utilised web3.js, a JavaScript library that enables interaction with Ethereum-based smart contracts. This library provided a simple and intuitive interface for users to interact with our smart contract-based incentivisation mechanism. With the help of these tools, we were able to implement a secure, reliable, and user-friendly IoT data sharing platform that incentivises users to contribute to scientific research.

4.D. SMART CONTRACT RECOMMENDATION ALGORITHM FOR IOT DATA MANAGEMENT AMONG STAKEHOLDERS

The smart contract for IoT data management covers the 1st step in Figure 1. At the beginning it defines a User struct that stores the user's preferences and wallet address. The preferences array stores the data types that the user is interested in. The authorizeUser and deauthorizeUser functions are used to control access to the contract. Only authorised users can set their preferences and retrieve recommendations.

The comprehensive workflow is available as a smart contract in our GitHub repository[7] and the *getRecommendation* function is developed as follows:

```
...
contract IoTDataManagement {
   IotDataInterface externalContract;
   struct User {
     uint256[] preferences;
     address wallet;
   }
mapping(uint256 => User) public users;
mapping(address => bool) public autho-
rizedUsers;
   IERC20 public token;

   constructor(IERC20 _token, address
_externalContract) {
token = _token;
externalContract=IotDataInterface(_ex-
ternalContract);
   }
...

   function getRecommendation(uint256
_userId) public view returns (uint256)
{
     require(authorizedUsers[msg.send-
er], "User not authorized");
     uint256[] memory userPreferences =
users[_userId].preferences;
     uint256 maxScore = 0;
     uint256 recommendedData = 0;
for (uint256 i = 0; i < userPreferenc-
es.length; i++) {
uint256 score = externalContract.get-
DatasetScore(userPreferences[i], to-
ken);

     if (score > maxScore) {
       maxScore = score;
   recommendedData = userPreferenc-
es[i];
     }
   }
   return recommendedData;
}
}
```

[2] https://github.com/JunkieLabs/sensify-android

[3] https://angular.io/

[4] https://www.mongodb.com/

[5] https://www.openzeppelin.com/

[6] https://web3js.readthedocs.io/

[7] https://github.com/sandig/IWSG-2023

The *setPreferences* function allows users to set their preferences by passing an array of data types. The *getRecommendation* function uses the preferences of a given user to calculate a score for each data type. The score is calculated from the external contract function *getDatasetScore* that may be further filtered into a specific domain based on the token. The data type with the highest score is recommended to the user. Overall, this smart contract provides an incentivised recommendation algorithm for IoT data management among stakeholders based on end-user requirements. Users are incentivised to contribute data types that are in demand, and the smart contract recommends the most valuable data type to each user. This helps ensure that data is collected and shared efficiently, benefiting both end-users and data providers.

5. EXPERIMENTAL EVALUATION

In this section we summarise the most resounding scientific findings.

5.A. SMART CONTRACTS FOR DATA PROVISION INCENTIVISATION

The incentivisation mechanism of the framework (see 2nd step in Figure 1) is fully automated through the smart contract, which ensures that the process is transparent and free from any bias or manipulation.

Deployment of the smart contract is performed by invoking the constructor function that initialises the data provider's address, the data consumer's address, the incentive amount and the state of the contract (Created). The *payIncentive* function can be called by the data consumer to pay the incentive amount to the data provider. The *getRefund* function can be called by the data provider to get a refund of the incentive amount. It checks if the sender is authorised to request a refund and if the state of the contract is Created. The incentivisation process may be additionally enhanced with an automated invocation of the functions through the smart oracles mechanisms (e.g. invocation of external APIs within the contract). This smart contract ensures that the data provider receives the incentive only when the data consumer pays the correct amount. It also allows the data provider to request a refund if the incentive is not paid. The described workflow is presented in the following smart contract.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

contract DataIncentive {
   address payable public dataProvider;
   address payable public dataConsumer;
   uint public incentiveAmount;
   enum State { Created, Paid }
   State public state;

   constructor(address payable _data-
Provider, address payable _dataConsum-
er, uint _incentiveAmount) {
     dataProvider = _dataProvider;
     dataConsumer = _dataConsumer;
     incentiveAmount = _incentiveA-
mount;
   state = State.Created;
   }

   function payIncentive() public pay-
able {
   require(msg.sender == dataConsumer
&& state == State.Created, "State or
access error");
 require(msg.value == incentiveAmount,
"Incorrect incentive amount");
     dataProvider.transfer(msg.value);
     state = State.Paid;
   }

   function getRefund() public {
   require(msg.sender == dataProvider
&& state == State.Created, "State or
access error");
     dataConsumer.transfer(incentiveA-
mount);
     state = State.Paid;
   }
}
```

5.B. QUALITATIVE ANALYSIS

The proposed incentivised recommendation framework for IoT data provision for scientific research incorporates various implicit blockchain properties that enhance the reliability and trustworthiness of the data. The transparency of the blockchain ensures that all transactions are visible and traceable, which is essential in the context of scientific research where the authenticity and accuracy of the data are critical. The tamper-proof nature of the blockchain guarantees that the data cannot be altered once it has been recorded, which prevents any fraudulent activities and enhances the trustworthiness of the data.

Furthermore, the decentralised nature of the node operators confirming transactions ensures that the data is not controlled by any centralised authority, making it trustless. Furthermore, privacy can be ensured with various cryptographic protocols on the Tangle ledger (e.g. L2Sec[x]) to keep secure and private sensitive data. Overall, the proposed incentivised recommendation framework for IoT data provision for scientific research incorporates various implicit blockchain properties, a generic recommendation algorithm, and a fully automated incentivisation mechanism. These features enhance the framework's reliability, trustworthiness, flexibility, and scalability, making it an ideal solution for incentivised data provision for scientific research.

The recommendation algorithm in the form of a smart contract, presented in the previous section, provides a generic recommendation algorithm suitable for all IoT subdomains. This enhances the flexibility and scalability of the framework, making it easily adaptable to different IoT subdomains. Moreover, the generic nature of the recommendation algorithm also ensures that it can be easily integrated into existing IoT systems without significant modifications, which is essential in scientific research, where interoperability and interdisciplinarity are important.

5.C. SUSTAINABILITY ANALYSIS

ShimmerEVM is feeless by design. In general, there are no fees for executing transactions, minting Non-Fungible Tokens (NFTs) or anchoring smart contract chains. In fact, these operations including their metadata is stored on the actual physical hardware of the node operators. Without limiting the storage, the node owners would be forced to add terabytes upon terabytes of additional storage space. For this reason, the storage is bound to the number of native tokens (SMR) in outputs, so the SMR tokens turn into its storage deposit. In case when the data exceeds the owned SMR tokens the network will deny the transaction. By refunding the storage the token is refunded.

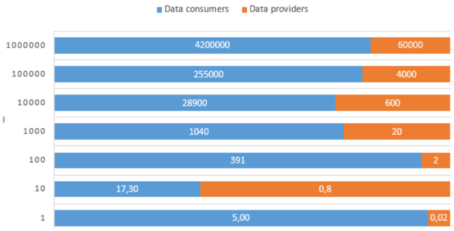


Figure 3: Sustainability analysis based on number of smart contracts deployed (1-10^6) with SRM tokens used between the users.

The sustainability analysis consists of an empirical analysis of the proposed DataIncentive smart contracts on ShimmerEVM testnet network deployed and triggered by the data consumers and data providers. The smart contract operations represent on-chain data incentive use case where we simulated the incentive smart contract use case 106 times. The results depicted in Figure 3 shows that the proportion of consumed SRM tokens converges to the higher percentage of usage to the data consumer within the increasing number of deployed smart contracts. Data providers are significantly less affected by contract operations. On the other hand, data consumers need to purchase on average less than 5 SMR per smart contract use case that is less than 0.35 USD for every iteration. In any case, both users should use the storage refunding approaches that are not in this paper's main scope.

6. DISCUSSION AND CONCLUSION

The proposed trustless incentivised recommendation framework for IoT data provision to data consumers (researchers) is an innovative solution that employs a blockchain-based on-chain recommendation framework to provision relevant IoT data to the researchers. DLT components enhance transparency, data integrity, and scalability of trustless agreements among system users, which are packed as smart contracts.

One of the key benefits of the proposed system is its ability to incentivise data providers to provide IoT data for research purposes, which can improve various domains such as smart cities, medical fields, and understanding people's habits. This can lead to significant advancements in these areas and ultimately help people in various ways.

The use of smart contracts enables the system to operate in a trustless environment, where all parties can agree on the terms and conditions of the data exchange without relying on a central authority or intermediaries. This provides a high level of transparency, immutability, and security for the data exchange, which is critical today, where data privacy and security are increasingly important. The empirical evaluation of the system in a testnet environment indicates that the proposed approach is sustainable and can provide reliable and efficient data provisioning for researchers. However, it is essential to note that the system needs further evaluation and testing in real-world scenarios to determine its effectiveness and scalability.

In conclusion, the proposed trustless incentivised recommendation framework for IoT data provision to data consumers is a promising solution that can revolutionise how researchers access and use IoT data. Blockchain technology and smart contracts can enhance transparency, data integrity, and scalability of trustless agreements among system users. This can lead to significant advancements in various domains, ultimately benefiting people in many ways.

7. ACKNOWLEDGMENT

The research and development reported in this paper received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 957338 (ONTOCHAIN: Trusted, traceable, and transparent ontological knowledge on the blockchain) and from the Research Agency of the Republic of Slovenia under the research programme P2-0426 Digital Transformation for Smart Public Governance 1/1/22-12/31/27.

8. REFERENCES

i J. Neeli and S. Patil, (2021). Insight to security paradigm, research trend & statistics in internet of things (IoT). Global Transitions Proceedings, 2(1), 84–90. https://doi.org/10.1016/j.gltp.2021.01.012

ii Maddikunta, P. K. R., Pham, Q. V., Nguyen, D. C., Huynh-The, T., Aouedi, O., Yenduri, G., Bhattacharya, S., and Gadekallu, T. R. (2022). Incentive techniques for the internet of things: a survey. Journal of Network and Computer Applications, 103464. https://doi.org/10.1016/j.jnca.2022.103464

iii Silvano, W. F. and Marcelino, R. (2020). Iota Tangle: A cryptocurrency to communicate Internet-of-Things data. Future Generation Computer Systems, 112, 307–319. https://doi.org/10.1016/j.future.2020.05.047

iv Nakanishi, R., Zhang, Y., Sasabe, M., and Kasahara, S. (2020). IOTA-based access control framework for the Internet of Things. 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Paris, France, 2020, 87–95. https://doi.org/10.1109/BRAINS49436.2020.9223293

v Sealey, N., Aijaz, A., and Holden, B. (2022). IOTA Tangle 2.0: Toward a scalable, decentralized, smart, and autonomous IoT ecosystem. 2022 International Conference on Smart Applications, Communications and Networking (SmartNets), Palapye, Botswana, 2022, 01–08. https://doi.org/10.1109/SmartNets55823.2022.9994016

vi Zhang, Y., Kasahara, S., Shen, Y., Jiang, X., and Wan, J. (2019). Smart contract-based access control for the Internet of Things. IEEE Internet of Things Journal, 6(2), 1594–1605. https://doi.org/10.1109/JIOT.2018.2847705

vii Gec, S., Stankovski, V., Lavbič, D., and Kochovski, P. (2023). A recommender system for robust smart contract template classification. Sensors, 23(2), 639. https://doi.org/10.3390/s23020639

viii Lone, A. H. and Naaz, R. (2021). Applicability of Blockchain smart contracts in securing Internet and IoT: A systematic literature review. Computer Science Review, 39, 100360. https://doi.org/10.1016/j.cosrev.2020.100360

ix Mihaylov, M., Razo-Zapata, I., and Nowé, A. (2018). NRGcoin—A blockchain-based reward mechanism for both production and consumption of renewable energy. In: Marke, A. (Ed.), Transforming climate finance and green investment with blockchains. Cambridge, MA, Academic Press, p. 111–131. https://doi.org/10.1016/B978-0-12-814447-3.00009-4

x Carelli, A., Palmieri, A., Vilei, A., Castanier, F., and Vesco, A. (2022). Enabling secure data exchange through the IOTA Tangle for IoT constrained devices. Sensors, 22(4), 1384. https://doi.org/10.3390/s22041384

STARTING WITH A FIRM FOUNDATION: BUILDING A SUSTAINABILITY PROGRAM FOR THE AUSTRALIAN RESEARCH DATA COMMONS

CLAIRE STIRM
San Diego Supercomputer Centre, University of California, San Diego, La Jolla, California, USA
cstirm@UCSD.EDU

KERRY LEVETT
Australian Research Data Commons, Adelaide, South Australia, AUS
kerry.levett@ardc.edu.au

NANCY MARON
BlueSky to BluePrint Consulting, New York, New York, USA
nancymaronconsulting@gmail.com

SANDRA GESING
Discovery Partner Institute, University of Illinois, Chicago, Illinois, USA
sgesing@uillinois.edu

ANDREW TRELOAR
Australian Research Data Commons, Melbourne, Victoria, AUS

SIOBHANN MCCAFFERTY
Australian Research Data Commons, Brisbane, Queensland, AUS
siobhann.mccafferty@ardc.edu.au

JULIANA CASAVAN
Casavan Consulting, West Lafayette, Indiana, USA
julianabcasavan@gmail.com

ABSTRACT

The United States Science Gateways Community Institute (SGCI) and the Australian Research Data Commons (ARDC) have offered their communities support and training around project sustainability. The support given has varied from in-person sustainability training, such as SGCI's Focus Week, direct advice giving, planning consultation, and personnel support to make sure the project continues to succeed within the traditional academic environment. All of these methods have helped research projects in the United States and in Australia.

In 2020, SGCI and ARDC initiated the planning of a Sustainability Program. The Sustainability Program was based on curriculum developed by the SGCI instructors, while being customized to cover core topics with follow-on cohort feedback sessions with the ARDC program staff. The Sustainability Program gave projects in-depth training on core strategies to »think like a business« while operating in an academic environment; technology best practices for science gateway user-interfaces; and long-term sustainability strategies to receive continued support.

Keywords: *Sustainability, Training, Science Gateways, Virtual Research Environments*

1. INTRODUCTION

Sustainability continues to be a widely discussed topic within academic research, especially in relationships with research software projects. Over 90% of researchers answering surveys say that they use software for their research and over 65% express that they even could not do their research without software[i]. With more and more projects dependent on research software and science gateways, sustainability

https://dx.doi.org/10.15496/publikation-100323

of such solutions is crucial to allow researchers to focus on their work instead of setting up or re-developing frameworks when availability of existing research software and science gateways disappear when financial or community support goes away[ii]. Sustainability of science gateways has many facets: from technical aspects such as good practices in software engineering to usability of science gateways to community building practices to attract a large community. Widely used science gateways have better chances to be further funded or supported, one further aspect of sustainability. Diversifying funding contributes to ongoing operations of science gateways and being able to develop additional features beneficial for the community.

Organisations such as the United States Science Gateways Community Institute (SGCI)[iii],[iv],[v] and the Australian Research Data Commons (ARDC)[vi] offer support and training to projects navigating sustainability challenges and potential paths towards a more secure future. In 2020, these two organisations formed a partnership to deliver a unified, customised sustainability program to projects supported by ARDC. In this paper we discuss the preparation and delivery of the ARDC Sustainability Program. Additionally, we present the outcomes of the training.

2. BACKGROUND

Researchers who build science gateways are often seeking to address, understand, and solve a specific problem. These challenges could pertain to urban development planning, monitoring fish populations, or certifying modelling tools for researchers. Curious researchers find inspiration in the challenge and seek to find a solution that positively impacts broader audiences, such as their fellow research peers, students, policy makers, or the general public. Once a solution has been conceptualised or produced in the form of a product or service, follow-on questions emerge.

- How can the project team define the value of their work in a way that inspires opportunities for continued support?
- How can the project reach intended audiences or even create change in a societal system?
- How can the project further develop within the means provided in an academic community?

These questions stem from a need for sustainability in order to continue the efforts of the project. Cyberinfrastructure projects combining access to data with software tools and underlying compute resources are commonly referred to as virtual research environments, science gateways, or research platforms. These projects straddle the line between computer science and hard sciences, as they bring together these two worlds to provide digital access to cutting-edge research models, educational softwares, or a central interactive data repository. Due to their blended nature, they need a team that has computer science and engineering, system administration, and research skills. They also have a continued need for software and hardware to enable consistent access for their audiences, while incorporating additional resources as needed when growth occurs. All of these needs add up. Often projects receive a lump sum of funding at the initiation of an idea yet struggle to receive a consistent form of funding without additional buy-in from new audiences or new support groups. Sometimes the sustainability challenge is not always monetarily driven but driven by a need for expertise in specific development practices or community engagement.

These discussions and more have emerged from peer workshop series such as WSSSPE (Workshop on Sustainable Software for Science: Practice and Experiences)[vii], on-campus conversations with researchers[viii], journals such as JORS (Journal of Open Research Software)[ix], and funding agencies such as the United States National Science Foundation (NSF)[x]. Other professional organisations carry forward these thought pieces such as the UK Software Sustainability Institute (SSI)[xi], the US Research Software Engineers Association (US-RSE)[xii], the consortium of Advanced Cyberinfrastructure-Research and Education Facilitators (ACI-REF)[xiii], and the United States Science Gateways Community Institute(SGCI)[xiv].

In August 2016, the Science Gateways Community Institute (SGCI) was created to provide subsidised services and resources to the developers and users of science gateways. During the first year of SGCI, founder Nancy Wilkins-Diehr, presented at the University of Queensland with an overview of SGCI's offerings and interest in collaborations including the International Coalition on Science Gateways and the International Workshop on Science Gateways[xv]. This foundation led to continued following and conversations between SGCI and ARDC. SGCI developed a science gateway sustainability training model, Focus Week, to train science gateway owners in all lifecycle stages of the core sustainability tools[xvi]. Three main areas are addressed in the training:

1. Core business strategy skills as they apply to leading an online digital presence, such as understanding stakeholder and user needs; business, operations, finance, and resource planning; marketing and project management.
2. Technology best practices, including the principles of usability and user-centred design for science gateway environments.
3. Long-term sustainability strategies, such as alternative funding models; case studies of successful gateway efforts; licensing choices and their impact on sustainability.

The training is set up for five days as an interactive workshop with a maximum number of ten teams to be accepted for each workshop. The teams consist ideally of project members in diverse roles such as the science gateway's Principal Investigator, project managers, lead research scientists, community managers, or developers. The first Focus Week, which was previously called Bootcamp, was held in April 2017. Ten teams were in attendance with one observer from Nectar Australia, the research cloud founded in 2013 and that has been applied by ARDC from 2018. This observation of the SGCI provided training had the initial goal to initiate similar conversations and training in Australia. This interaction led to a partnership forming between Nectar and SGCI. The feedback of the participants showed that the concept was found beneficial, thought-provoking and entertaining with some room for improvement.

Past the initial delivery of Focus Week, SGCI has continued to provide the United States and international communities with sustainability training sessions, largely held as in-person workshops. Twenty-one training sessions have been delivered by SGCI Instructors to over 670 participants that work on science gateways or research projects. In one example, a shortened sustainability course was delivered to attendees at the 2018 International Workshop for Science Gateways[xvii]. In December 2020, SGCI delivered the Focus Week sustainability training as a two-week virtual course to 57 attendees. While the majority were project teams from the United States, there were attendees joining internationally including staff members from the Australian Data Research Commons (ARDC).

The ARDC is funded by the Australian Government's National Collaborative Research Infrastructure Strategy and is designed to accelerate Australian research and innovation by driving excellence in the creation, analysis and retention of high-quality data assets[xviii]. The ARDC partners with the research community and industry to build leading-edge digital research infrastructure and runs a series of programs including the ARDC Platforms Program.

The ARDC Platforms Program seeks to enable transformative research across all disciplines using advanced software and platforms and supports 26 projects that are building platform infrastructure (i.e. virtual research environments) between 2019-2023. After participating in the December 2020 Focus Week, the ARDC team partnered with the SGCI Focus Week instructors to design and deliver a sustainability training program that would meet the needs of project teams supported by the ARDC.

3. SGCI FOCUS WEEK CURRICULUM

The SGCI Focus Week is an intensive workshop designed for innovative research teams to work together on producing a strong sustainability plan[xix],[xx],[xxi]. Participating projects leave Focus Week with a clearer definition of their project's value, its audience, and its positioning in the competitive landscape. Below are the core Focus Week exercises that teams complete.

- Napkin Drawing: Learn how to effectively communicate your project.
- Understanding Your Audiences & Key Stakeholders: Identify who cares about your project, and determine why they care; explore potential new user groups.
- Environment-Mapping the Landscape: Spend time researching and mapping out your known and new competition, as well as open opportunities where your project is applicable.
- Marketing Tactics and Tools: Learn how to communicate your project's value to your audiences with selective marketing.

- Goal Setting: Think about the big picture ideas. Learn how you can set the right goals for the right reasons, and learn how you can measure your success.
- Value Proposition: Build a concise value statement that articulates the unique value your project delivers to its users.
- Budgeting: Discover how you can forecast a budget that will help you plan for life beyond the grant.
- Market Development: Explore the possible customer groups and subgroups that will find value in your project other than the original audience your project is intended to serve.
- User-Centred Design: Learn from a usability expert on best practices when designing cyberinfrastructure user interfaces.

The last day of the workshop is a »Pitch Day« with each team presenting their sustainability plans. This is a very rewarding experience for not only the presenting teams but their fellow cohort members as they can see how each other have grown upon each exercise and the outcome of putting all the exercises into one conclusive presentation. Many of SGCI's past attendees have shared their experiences from attending sustainability training sessions in the SGCI storybook[xxii]. Here are a few notable quotes from past participants:

> »ESIP Lab has funded over 20 projects since we attended Focus Week, and I'm able to take the tools that I learned and just use that language and pass it on to those projects. It gave me a mindset to continually evaluate and re-evaluate the Lab, too—Is our value proposition the same? What's our niche?«—Annie Burgess, ESIP Lab
>
> »We're so grateful for Focus Week because it is exactly the right thing for people who want to expand, broaden, and capitalise on their gateways. You can't do any of that without the training and resources provided by Focus Week. It was such an eye-opening experience for us and it remains, behind the scenes, what keeps us from going over a cliff.«—Jason Fleming, Coastal Emergency Risks Assessment Tool.

4. RE-IMAGINING FOCUS WEEK FOR ARDC

Through attending the virtual Focus Week in 2020 hosted by SGCI, ARDC staff started to explore a collaboration: a jointly run sustainability program for ARDC projects. Taking the Focus Week curriculum as a starting point, the SGCI team was asked to modify and customise the program to suit ARDC project leads. Rather than a one-week, in-person intensive, could Focus Week be presented as a series of virtual, interactive events, delivered over time? Working together, the SGCI team met with the ARDC team to develop a plan that would provide as much content as possible, with teams many time zones apart.

To address the time-zone differences, the teams determined that evening sessions in the US would work as morning sessions for Australian participants, and to address Zoom-fatigue, it was suggested (and eagerly adopted) that no sessions run longer than 90 minutes. A major benefit of in-person delivery of the Focus Week program—both to instructors and to participants—has been the ability to interact with project leaders after each topic has been discussed, and the ability for participants to take the time to work through the new material, and apply what they are learning to their own projects. To build in feedback while using a virtual delivery approach, the ARDC program was re-structured:

- First, SGCI instructors would present a core topic, in a virtual session, encouraging interactivity as much as possible.
- Each day's presentation was then immediately followed by an SGCI and ARDC debriefing session, permitting the SGCI instructors to meet with ARDC program leads to review the material covered, and prepare for the work sessions to come
- A formal »work session« for participants was then led by ARDC staff to provide feedback to project teams and answer questions.

The curriculum was customised to cover core topics with follow-on cohort feedback sessions with the Platforms program staff after each Tuesday and Wednesday session – see Table 1 for details.

An important element that was incorporated into the ARDC Sustainability Program was time for feedback after core training sessions. These feedback ses-

Date	Core Topics (90 Minutes)	Cohort Feedback (60 Minutes)
21-Feb	Introduction to sustainability. Napkin Draw-Ing: effectively communicate the value of your project through verbal and visual communication.	None
22-Feb	Audience: explore and assess your audiences and stakeholders to better engage them.	Team Breakouts, feedback to teams from ardc platforms team
23-Feb	Landscape: define who your competitors/potential collaborators are and how you differentiate your product from theirs	Team Breakouts, feedback to teams from ardc platforms team
24-Feb	Value proposition: identify the primary value that your project brings to its users and community	None
3-Mar	Goal setting & budgeting (2h): use impact driven goals to begin developing a budget and financial forecast	None
9-Mar	Revenue models: exploration into different revenue types that can work within the academic world	Team Breakouts, feedback to teams from ardc platforms team
17-Mar	Sales and marketing: learn principles of sales and marketing, and how to develop a plan to connect with your target audience and promote your gateway.	Team Breakouts, feedback to teams from ardc platforms team
23-Mar	Sustainability strategy »pitch«: each team presents the exercises they completed during the workshop in the form of a »pitch«	None

Table 1: ARDC Sustainability Program timeline

sions were scheduled to allow projects to have time with the ARDC staff members to share their concepts from the core topic exercise and discuss. This feedback allowed the projects to share ideas with their support team and it provided ARDC staff the opportunity to help advise on workable approaches for next steps.

5. ARDC SUSTAINABILITY PROGRAM DELIVERY

The ARDC Sustainability Program was held over the course of two months. The first part of the program was delivered over the course of one week, Monday, February 21, 2022 through Thursday, February 24, 2022. The sessions delivered are described in Table 1: Part 1. The goal of delivering all of these sessions in one concurrent week was to lay the foundation for sustainability practices and build enthusiasm for the concepts. The second part of the program was delivered as individual sessions once a week, described in Table 1: Part 2. The second part of the program was built on top of the foundation week with time in between sessions to allow for research and deeper team conversations to take place. The ending session was »Pitch Day«. A total of 74 attendees participated in the ARDC Sustainability Program. Additionally, 16 projects delivered their completed »Pitch Day« presentations at the end of the program.

6. FEEDBACK FROM ARDC SUSTAINABILITY PROGRAM PARTICIPANTS

At the end of the program, participants expressed to ARDC staff that they found the program highly valuable. With the goal of capturing some of this feedback, an exit survey was sent to the teams that attended the entire program.

An open text question at the end of the survey asked respondents to »share any additional comments, questions, or concerns.« Three survey participants shared:

- »Our platform made a lot of valuable progress in these sessions and spent a lot of time outside of the sessions to work on the activities. We ended up with a better understanding of our platform. I feel like there should be more focused workshops like these for platforms in the future.«
- »It was much more useful than I expected it to be. I think what our team has learnt will go a long way towards helping create a sustainable platform. Thanks very much!«
- »Great initiative!«

One survey participant also pointed out the value in being able to meet with the ARDC team during program feedback sessions.

> »I think the afternoon sessions where teams were able to meet and work on the exercises were really valuable. We had a breakthrough in one session when we locked in our value propositions and another when we identified the two axes of our market landscape. We definitely needed the time in these sessions to have those discussions.«

Participants were asked in the final survey, »How important do you think the following components are to the success of your project?«. There were ten curriculum items they were asked to rate with the options of extremely important, very important, moderately important, slightly important, and not at all important. The following response was provided to this question:

- Basics of sustainability strategy
  -100% of survey participants rated this as »Extremely important«
- Understanding your audience
  -100% of survey participants rated this as »Extremely important«
- Competitive landscape
  -50% of survey participants rated this as »Extremely important«
  -50% of survey participants rated this as »Very important«
- Value Proposition
  -75% of survey participants rated this as »Extremely important«
  -25% of survey participants rated this as »Slightly important«
- Market development
  -75% of survey participants rated this as »Extremely important«
  -25% of survey participants rated this as »Slightly important«
- Goal setting
  -25% of survey participants rated this as »Extremely important«
  -75% of survey participants rated this as »Very important«
- Budgeting
  -75% of survey participants rated this as »Extremely important«
  -25% of survey participants rated this as »Very important«
- Revenue Models compatible with Open Education
  -25% of survey participants rated this as »Extremely important«
  -75% of survey participants rated this as »Very important«
- Sales / Marketing
  -50% of survey participants rated this as »Extremely important«
  -50% of survey participants rated this as »Very important«
- Delivering a pitch
  -50% of survey participants rated this as »Extremely important«
  -25% of survey participants rated this as »Very important«
  -25% of survey participants rated this as »Slightly important«

Participants were also asked, »How would you rate the length of the series« in which the choices were way too long, a bit long, just right, a bit too short, and way too short. To this question, 75% of participants shared that it was the right length and 25% of participants found the series to be a bit too short. Finally, participants were asked, »Overall, how well did the series meet your expectations« in which the choices were extremely well, very well, moderately well, slightly well, and not well at all. Survey participants responded with 50% saying extremely well, 25% very well, and 25% slightly well.

The incorporation of feedback went past the initial program delivery. Based on the SGCI's practice of holding follow-on sessions to connect with teams after the sustainability program, the ARDC team scheduled a three-month follow-on session to meet with projects and hear updates on goals teams had set for themselves. Platform teams met the majority of their three-month goals, particularly those that had set goals specific to investigating or implementing sustainability activities such as engaging effectively with their audiences or developing operational budgets.

Both the SGCI and ARDC teams have viewed this collaboration as a success. ARDC's project Steering Committee noted that this work permitted project teams to begin looking beyond the time-bound project funding and duration, and develop the knowledge required for projects to become ongoing operational infrastructure. Sustainability is now considered a standing agenda item for them. ARDC staff stated that it was beneficial for them to participate as well as they heard the same language as the participating projects and could reflect on ARDC practices through this sustainability lens.

7. CONCLUSION

The ARDC Sustainability Program provided an opportunity to deliver sustainability training to projects supported by ARDC which benefited teams gearing up to initiate sustainable paths to continue supporting their project's goals. Additionally, this opportunity provided ARDC and SGCI the space to partner on a larger shared vision, to provide education on how projects can continue past initial support, and to have a shared space to discuss what it means for a project to be sustainable with project teams.

As the ARDC develops its Thematic Research Data Commons (RDCs), it is taking key learnings from the course into the design of new programs. Future projects supported by the ARDC will be supported to plan for sustainability from the beginning of the project by, for example, creating activity-based budgets, and engaging more deeply with their user communities.

As SGCI funding from NSF has begun to conclude, a new Center of Excellence has been awarded by NSF to continue providing sustainability training and other services to the science gateways community. Called the Center of Excellence to Extend Access, Expand the Community, and Exemplify Good Practices for CI through Science Gateways (SGX3), the science gateways community will continue to be offered sustainability training through in-person Focus Week workshops and virtual Jumpstart short courses[xxiii].

8. ACKNOWLEDGMENT

The Sustainability Program was initiated and supported by the Australian Research Data Commons. Additionally, the Science Gateways Community Institute was involved with the development of the curriculum and delivery of the initial ARDC Sustainability Program. The instructors of the initial ARDC Sustainability Program are from BlueSky to BluePrint, Casavan Consulting, Purdue University, and the San Diego Supercomputer Center at the University of California, San Diego. The Science Gateways Community Institute was originally funded by the U.S. National Science Foundation (ACI-1547611).

9. REFERENCES

i Katz, D. S., Niemeyer, K. E., Gesing, S., Hwang, L., Bangerth, W., Hettrick, S., Idaszak, R., Salac, J., Chue Hong, N., Corrales, S. N., Allen, A., Geiger, R. S., Miller, J., Chen, E., Dubey, A., and Lago, P. (2017). Report on the Fourth Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE4), arXiv:1602.02296 [cs.SE]

ii Nangia, U., and Katz, D. (2017). Survey of National Postdoctoral Association—Dataset [Data set]. Zenodo. https://doi.org/10.5281/zenodo.84360

iii Wilkins-Diehr, N., and Crawford, T. (2018). NSF's Inaugural Software Institutes: The Science Gateways Community Institute and the Molecular Sciences Software Institute.In: Computing in Science Engineering, vol. 20, no. 5, pp. 26–38, Sep./Oct. 2018. https://doi.org/10.1109/MCSE.2018.05329813

iv Wilkins-Diehr, N., Zentner, M., Pierce, M., Maytal Dahan, Lawrence, K., Hayden, L., and Mullinix, N. (2018). The Science Gateways Community Institute at Two Years. In: Proceedings of the Practice and Experience on Advanced Research Computing (PEARC '18). Association for Computing Machinery, New York, NY, USA, Article 53, 1–8. https://doi.org/10.1145/3219104.3219142

v Gesing, S., Wilkins-Diehr, N., Dahan, M., Lawrence, K., Zentner, M., Pierce, M., Hayden, L., and Marry, S. (2017). Science Gateways: The Long Road to the Birth of an Institute. In: Proceedings of HICSS-50 (50th Hawaii International Conference on System Sciences), 4–7 January 2017, Hilton Waikoloa, HI, USA, pp. 6243–6252. https://hdl.handle.net/10125/41919

vi Australian Research Data Commons—ARDC. (2023). Retrieved April 4, 2023, from https://ardc.edu.au/

vii Katz, D. S., Niemeyer, K. E., Gesing, S., Hwang, L., Bangerth, W., Hettrick, S., Idaszak, R., Salac, J., Chue Hong, N., Corrales, S. N., Allen, A., Geiger, R. S., Miller, J., Chen, E., Dubey, A., and Lago, P. (2017). In: Report on the Fourth Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE4), arXiv:1602.02296 [cs.SE]

viii Gesing, S., Lawrence, K., Dahan, M., Pierce, M. E., Wilkins-Diehr, N., and Zentner, M. (2019). Science gateways: Sustainability via on-campus teams, Future Generation Computer Systems, vol. 94, pp. 97–102. https://doi.org/10.1016/j.future.2018.09.067

ix Journal of Open Research Software. (2023). Retrieved April 4, 2023, from https://openresearchsoftware.metajnl.com/

x National Science Foundation—NSF. (2023). Retrieved April 4, 2023, from https://www.nsf.gov/

xi The Software Sustainability Institute. (2023). Retrieved April 4, 2023, from https://www.software.ac.uk/

xii The United States Research Software Engineer Association. (2023). Retrieved April 4, 2023, from https://us-rse.org/

xiii Aci-Ref. (2023). Ref. ACI. Retrieved April 4, 2023, from https://aci-ref.github.io/

xiv Gesing, S., Dahan, M., Zentner, M., Wilkins-Diehr, N., and Lawrence, K. (2019). The Science Gateways Community Institute: Collaborations and efforts on international scale, Future Generation Computer Systems, vol. 101, pp. 951–958. https://doi.org/10.1016/j.future.2019.07.024

xv Wilkins-Diehr, N. (2016). The Science Gateways Community Institute. University of Queensland Research Computing Centre Seminar Series. Retrieved April 4, 2023, from https://rcc.uq.edu.au/event/737/science-gateways-community-institute

xvi Gesing, S., Zentner, M., Casavan, J., Hillery, B., Vorvoreanu, M., … and Maron, N. (2017). Science Gateways Incubator: Software Sustainability Meets Community Needs. In: 2017 IEEE 13th International Conference on e-Science (e-Science), Auckland, New Zealand, 2017, pp. 477–485. https://doi.org/10.1109/eScience.2017.77

xvii IWSG 2018. (2023). IWSG 2018 Program. Retrieved April 4, 2023, from https://sites.google.com/a/nd.edu/iwsg2018/program

xviii Australian Research Data Commons—ARDC. (2023). Retrieved April 4, 2023, from https://ardc.edu.au/

xix Stirm, C., Zentner, M., Casavan, J., Hoebelheinrich, N., Craddock, R.C., Cleveland, S., Gesing, S., Eschrich, A-M. (2018). Science Gateways Community Institute Incubator Pitch Deck: Success Stories from the 2nd & 3rd Bootcamp. https://doi.org/10.6084/m9.figshare.7523330.v1

xx Gesing, S., Zentner, M., Casavan, J., Hillery, B., Vorvoreanu, M., Heiland, R., Marru, S., Pierce, M., Mullinix, N., and Maron, N., (2017). Science Gateways Bootcamp: Strategies for Developing, Operating and Sustaining Science Gateways. https://www.slideshare.net/slideshow/sgci-science-gateways-bootcamp-strategies-for-developing-operating-and-sustaining-science-gateways/80960871. Retrieved August 27, 2024

xxi Maron, N. (n.d.). Publications. BlueSky to BluePrint. Retrieved April 4, 2023, from https://www.blueskytoblueprint.com/publications

xxii Science Gateways Community Institute. (2022). SGCI storybook Connecting people, creating solutions, accelerating discovery. sciencegateways. Retrieved April 4, 2023, from https://sciencegateways.org/about/storybook

xxiii SGX3. (2023). Center of Excellence to Extend Access, Expand the Community, and Exemplify Good Practices for CI through Science Gateways. sciencegateways. Retrieved April 4, 2023, from https://sciencegateways.org/

FROM DATAPLANT'S DATAHUB TO DATAPUB(LICATION)

JONATHAN BAUER*, MARCEL TSCHÖPE, DIRK VON SUCHODOLETZ,
CRISTINA MARTINS RODRIGUES, JULIAN WEIDHASE
University of Freiburg, Freiburg i. Br., Germany
*jonathan.bauer@rz.uni-freiburg.de

TIMO MÜHLHAUS, CHRISTOPH GARTH, GAJENDRA DONIPARTHI
RPTU University Kaiserslautern, Kaiserslautern, Germany

HOLGER GAUZA, LOUISA PERELO
University Of Tübingen, Tübingen, Germany

ABSTRACT

A core objective of the DataPLANT consortium is to provide a science gateway as a technical basis that offers software engineering-inspired approaches for data management and makes them accessible to plant researchers. We are presenting the DataPLANT DataHUB which provides various RDM workflows to support research data scientists in different phases of the data life cycle—from the annotation and structuring of gathered data to the publication of the results obtained.

Keywords: *FAIR Data Sharing; Science Gateways; RDM Platform; Data Versioning; Data Publication; Workflow Integration*

1. MOTIVATION

Scientists in fundamental plant research increasingly rely on services supporting them to collaboratively manage their research data. In this sub-field of biology, the (molecular) principles of plant life are investigated, which determine plant growth, crop yield and biomass production. Methods used by different groups in the field include transcriptomics, proteomics and metabolomics. Especially cross-disciplinary and -institutional collaboration as well as the use of data of different modalities—from many sources and experiments, pre-processed or analyzed with a variety of algorithms—requires contextualization. Scientists need a common base to exchange and understand each other's data and steps in data analysis. To cooperate beyond local labs where files are traditionally exchanged using typical file shares, suitable infrastructures need to be established to assist research data collaboration. A well-adapted shared research infrastructure fosters the collection, processing, exchange, citation and archiving of research data and its contexts.

Our paper extends upon the concepts and ideas of »Immutable yet evolving: ARCs for permanent sharing in the research data-time continuum«[i] and gives an overview on the results of the developments in the DataPLANT consortium[ii] so far to provide a domain specific science gateway. This gateway consists of a common infrastructure to be used to share, version, exchange and publish research data in a transparent and open fashion. We transfer concepts and workflows used for years in software development to the domain of experimental science[iii]. The science gateway establishes a central entry point to valuable subject-specific data and domain-specific knowledge. The DataHUB service environment provides the necessary means to contextualize research data according to the FAIR principles with minimal additional effort, and to support the entire research cycle in modern plant biology. Researchers are neither data managers nor IT experts and require practical assistance in exploiting the up to now fragmented and complex resource landscape.

DataPLANT is part of the cross-domain German National Research Data infrastructure (NFDI) inaugurated in late 2021[iv]. The NFDI creates a platform to develop research data management (RDM) services and infrastructures to advance data management in general and to improve cross-disciplinary exchange of data and information, as well as driving the digital transformation and democratization of research data[v]. Establishing RDM within the NFDI enables the combination of interdisciplinary exper-

https://dx.doi.org/10.15496/publikation-100323

tise, as well as the comparison and the integration of various analysis results.

2. A SCIENCE GATEWAY FOSTERING DATA-CENTRIC RDM

A central motivation in DataPLANT is to define and standardize easy-to-use RDM procedures and their technical realization, specifically targeted at the needs of the fundamental plant research community. DataPLANT considers FAIR digital objects (FDO) as the core of all relevant concepts and developments. To follow the proposed data centric approach for RDM, we specified the Annotated Research Context (ARC) that captures and structures the complete research cycle meeting the FAIR requirements with low friction for the individual researcher[vi]. ARCs are self-contained and include assay/measurement data, workflows and computation results accompanied by metadata in one FDO. The ARC structure allows full user control over all metadata and facilitates usability, access, publication and sharing of the research.

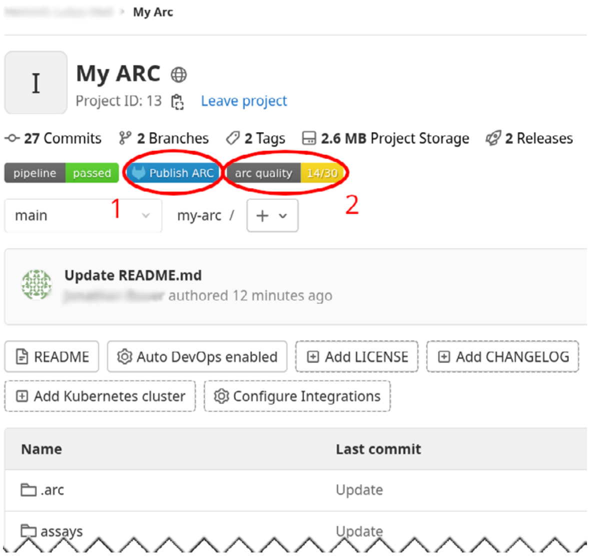
The DataPLANT DataHUB provides a central hosting facility for ARCs which are Git backed repositories, open to all researchers participating in the consortium and their collaborators. Having a platform in place for versioning and sharing as well as provenance tracking further reduces the burden of initiating collaboration with peers. Thus, the DataHUB acts in the role of a science gateway for plant researchers sustaining collaborative work and allowing sustainable data management.

3. BUILDING BLOCKS OF THE DATAPLANT DATAHUB

The development principles in DataPLANT are designed for reproducibility and (re)deployability in the central infrastructure of the hosting institutions as well as on-premises solutions and possibly later in a consolidated common NFDI base infrastructure. This allows for institutions, groups, or projects in fundamental plant research to use the DataPLANT service landscape harnessing their own resources, e.g., for sensitive data. We crafted the DataPLANT services deployment and update procedures in such a way to be used for both the consortium global instance and for on-premises solutions, to offer a common technological basis.

The DataHUB as a science gateway—the entry point to various services (starting with a versioned, generated web page)—is made accessible through a stable (highly available) entry point. On the technical side, this is a reverse proxy that integrates all the user-facing services independently of their physical location. The DataPLANT service infrastructure is cloud-enabled and relies on a common stable entry proxy handling DNS, TLS/SSL certificates and forwarding.

By design, services should be atomic and modularized to facilitate their integration, deployment, reusability, and extension. Services are automatically (re)deployable to enable cloud-based operation, while allowing distribution and adjustments to local requirements for on-premises setups. They follow the »infrastructure-as-code« principle wherever possible. The proxy makes it easy to allow parallel instances of services to test new versions before switching them in production. It also provides easy access to (temporary) prototypical services.



Name	Last commit
.arc	Update
assays	Update

Figure 1: The science gateway helps users to deal with their daily tasks on data management, metadata annotation, workflow execution or data publication in collaborations.

3.A. SECURE DATA HANDLING

Key features of the DataHUB are versioning, the possibility to work in groups, the support for multiple contributions and easy-to-use access management.

We found these key features in the GitLab project that we are customizing to the needs of the DataPLANT community. Versioning is provided by Git. Each file in a Git repository is tracked, so that changes can be undone. Users can revert any changes, primarily through the tools provided by DataPLANT in the form of the easy-to-use ARC Commander abstracting from the low-level Git commands[vii]. Of course, they can alternatively use GitLab's user interface or even fall back to the command line. Furthermore, files are also protected from accidental deletions and can easily be restored to previous versions. Another important feature is the branching mechanism of Git. This feature enables users to fork ARC repositories on their own, modify them, and eventually merge them again with the original ARC repository. GitLab also provides a fine-grained access management, to form groups for collaborations that the users can manage themselves. On the technical side, we are using GitLab's builtin LFS server to store the often very large data objects stored in ARCs[1]. The LFS files are stored in an object store via S3 in the backend.

The DataHUB platform is where ARCs are evolving until a specific state is attained. While these can be tagged or released, the platform is not meant to provide long-term access or citability. To meet these essential FAIR criteria, we deployed a data publication service complementing the DataHUB. It is realized with the InvenioRDM turn-key repository framework supported by a large international community of research institutions and led by CERN in Switzerland[2].

3.B. PRIMARY USER INTERFACE OF THE DATAHUB

The DataHUB user interface customizations focus on user-friendliness to lower the barrier of entry for researchers unfamiliar with software development and code versioning concepts. The default landing page is configured to show a list of public ARCs. New users can thus quickly discover relevant research data and explore the ARC concept. An ARC project template containing the basic directory and file structure of an ARC can be imported when creating new projects. Additional customizations on top of GitLab's CI/CD features have been made to automatically give the users feedback on the quality of their ARC using badges (Figure 1) and, potentially, initiate the publication.

3.C. GITLAB DEPLOYMENT AND CUSTOMIZATION

We needed to adapt GitLab to the requirements of the ARC concept by creating a modified GitLab Docker image. This customized Docker image includes changes to the user interface and the addition of special templates that allow users to easily create an ARC in the user interface. This Docker image is distributed via the GitHub Docker Registry for simple deployment and installation and can be used as an easy way to roll out DataHUBs on-premises[3].

For our central DataHUB installation, we provide a separate Docker image with slightly different modifications. This image additionally includes the InvenioRDM publishing mechanism, which allows the user to publish an ARC directly from the DataHUB.

We discussed several options for the publication process. The first option we evaluated revolved around using GitLab's built-in release mechanism. Releases are easy to use and generate an archive of the repository automatically. However, this mechanism has the disadvantage that LFS objects are contained in the archive and duplicates the LFS objects, which are usually very large files. These can be raw and/or temporary research data that are not always meant to be included in a publication. Another option would be to trigger the publication release directly within the Auto DevOps pipeline. In this case, the last pipeline step for publication can be designed to require user action. However, this method has the disadvantage that API credentials from InvenioRDM must be included in the pipeline scripts. But again, these credentials cannot be securely hidden from users.

For these reasons we decided to follow an ap-

[1] GitLFS is an open source extension that allows pointing to large files external to a GitLab repository. Thus, GitLFS reduces the size of a repository and helps managing large files. See https://git-lfs.com/ for further information.

[2] Project website: https://inveniosoftware.org/products/rdm/

[3] DataHUB repository: https://github.com/nfdi4plants/datahub

proach based on the event hooking mechanism of GitLab and a modified Auto DevOps pipeline. We favor this solution instead of project-specific CI/CD templates as they work out of the box and do not need to be set up by the users. The modified Auto DevOps pipeline checks the ARC directory and file structure for completeness and correctness and runs unit tests. After that, we generate badge icons as easy-to-understand feedback for the users (Figure 1, 2). In addition, users can utilize the test report view integrated in GitLab to view individual tests in detail. The publication metadata for an ARC is generated during this ARC validation step. This metadata is later used for the automatic publications in InvenioRDM. The complete pipeline is shown in Figure 3.

From the researchers' perspective, publishing an ARC from the DataHUB should be as easy as possible. For this purpose, a microservice automatically sets up a publish badge button (Figure 1, 1) for each newly launched project. The microservice encodes a token in the URL of the publish badge button. This token is encoded according to JWT (JSON Web Token) specifications. The token consists of a header, a payload, and a signature. The header indicates the algorithm used. The payload encodes the project ID and the project name. Finally, the header and payload are signed with an HMAC signature to verify their validity. This process ensures that only authorized people can publish ARCs. When a project needs to be published, users can simply click on the button. This will take them to the microservice's frontend. The frontend gives users a further overview of tests, metadata, and project information. Once all the data is available, users can publish the ARC.

In a first step, the microservice in the backend receives the previously generated test report from the ARC. If the test report and the pipeline are successful, the microservice continues.

In the case of success, the microservice retrieves the latest archive file and metadata from the artifacts and uploads the archive to the InvenioRDM instance. The user is finally informed about the success on the website frontend. Information about the user, the project and the details of the test report are retrieved using the GitLab REST API from the project ID which is contained in the JWT token. Since we are not publishing an ARC directly, e.g. a data steward or leading principal investigator has to review the publication and to accept it in InvenioRDM. Figure 4 illustrates the procedure in detail.

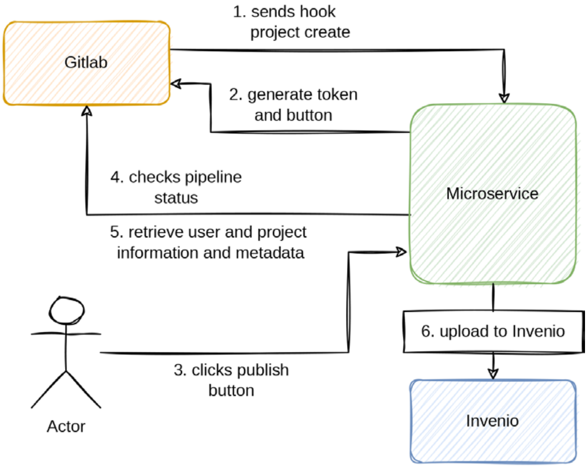

1. sends hook project create
Gitlab
2. generate token and button
4. checks pipeline status
Microservice
5. retrieve user and project information and metadata
6. upload to Invenio
3. clicks publish button
Actor
Invenio

Figure 2: Architecture: When a new project is created, Gitlab sends a system hook (1). This is received by the microservice, which then adds the publish button to the newly created project. A JWT token is inserted into the URL of the publish button for later authentication (2). When a user clicks on the publish button (3), it is passed to the microservice frontend. The microservice checks the pipeline of the project (4) and then receives more metadata (5). If the pipeline is successful, the user can publish the project and the microservice will publish the project to Invenio (6).

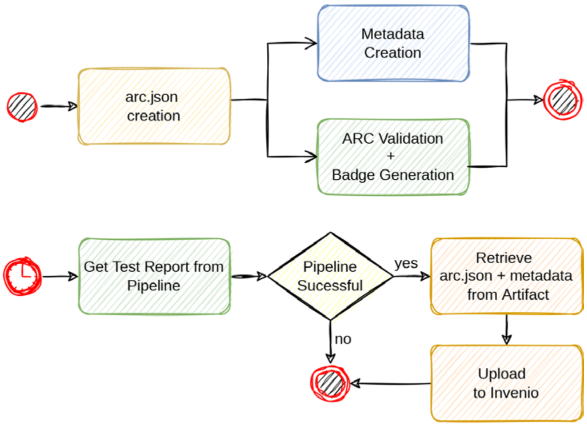

Metadata Creation
arc.json creation
ARC Validation + Badge Generation
Get Test Report from Pipeline
Pipeline Sucessful
yes
no
Retrieve arc.json + metadata from Artifact
Upload to Invenio

Figure 3: Trigger Microservice: When a user performs a release, the microservice is triggered. The microservice uploads the release to InvenioRDM if the pipeline was successful. The user is informed about the status by mail.

The microservice uses GitLab's event hook mechanism and therefore sets up a system hook to receive project creation events. The microservice installation is based on Docker. This allows for easy installation with little setup effort. The installation only requires the creation of a GitLab token with API access rights. The rest of the installation is automatic but can be controlled via a REST API. As this method is independent of our modified GitLab version, the microservice can be updated more easily.

The use of the microservice depends on proper authentication. Users are authenticated using the JWT token as described above. The microservice also requires access to both the InvenioRDM instance and a Gitlab API access. For both systems, access tokens must be passed to the microservice during installation. Nevertheless, both tokens are securely protected from external access. Since a new publish token must also be created when a new project is created, system hooks are used, which in turn are secured by a webhook secret. In addition, all requests are secured using TLS connections.

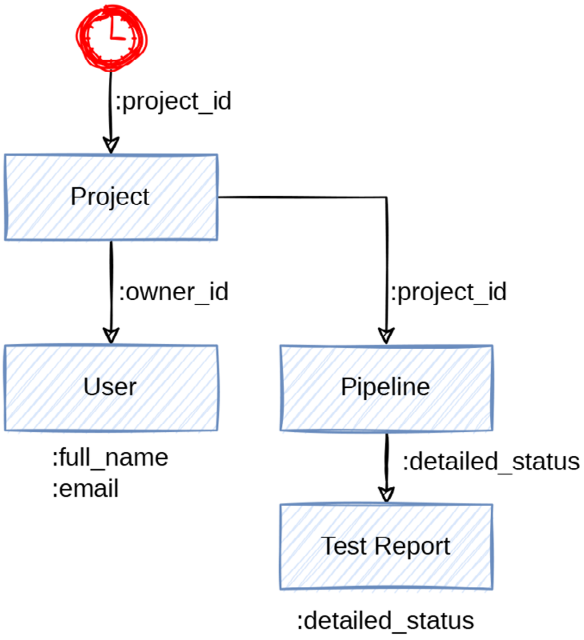


Figure 4: GitLab API Requests: When the microservice is triggered, pipeline status, test reports and user information are retrieved via the GitLab API. User information can be retrieved via a project's owner ID, while test reports can be retrieved via a detailed status query.

3.D. COMMON AAIs: LIFE SCIENCES AND ORCID

An authentication instance is required for authenticating users and the services behind them. The DataPLANT user management builds upon existing AAIs. Well established services like Life Sciences AAI and ORCID can be combined with local authentication within the central DataPLANT authentication service. The infrastructure leverages on Keycloak[4], developed by RedHat, that supports modern authentication protocols like OpenID-Connect and SAML and allows the integration of multiple AAIs and identity brokering. Providing an AAI identity management, which can easily be connected with GitLab and other services through either protocol, simplifies the user management. The connection of multiple AAIs through Keycloak enables our community to use their existing accounts, for example from the Life Sciences AAI, their home institution or ORCID. We can assign different roles depending on the account source or on specific attributes. Permissions can be derived from these roles to differentiate between users. These range from privileged users having full access to the data and the ability to create archives/publications, to underprivileged users that have only a reporting function and/or read-only access to raw data. All this is still in a very early stage and needs to be refined from more feedback on the productive use of the infrastructure.

3.E. INVENIORDM AS REFERENCE BROKER AND LONG-TERM ACCESS

The component in the DataPLANT's DataHUB to provide permanent references to published versions of ARCs[viii] in various forms of annotated research data, workflows and results is taken care of by InvenioRDM. It provides all the relevant interfaces to interact with other DataHUB components or external services. Core technical aspects are the availability of both a web user interface and a REST API, extensibility with respect to the integration of metadata schemas and vocabularies for the annotation of research data, and flexibility with respect to usable storage technologies. In this context, especially the support of object storage via S3 is future-proof and scalable.

The multi-tenancy of InvenioRDM enables the creation of scientific communities including the integration of workflows for quality assurance in the peer review process by dedicated members of this community. This feature alone allowed us to design the automated publication process in a user-friendly way. All ARC publications triggered from the DataHUB are automatically submitted for review in the community and must be signed off by the designated data steward/curator. Without this additional step, an automated publication workflow would likely lead to erroneous publications triggered by mistake, either

[4] Project website: https://www.keycloak.org

by the users themselves or by some automation, including unneeded DOI minting. In our approach, those can simply be discarded.

The integration of DataCite's API not only enables the assignment of DOIs but also DataCite's automatic profile update. Scientists can connect their ORCID account with DataCite. Upon new publications, DataCite will automatically transfer the data publication information to the user's ORCID profile. Using the OAI-PMH and REST API interfaces, other research information systems can systematically collect and reference the published datasets. An integration into Re2DATA.org and other harvesters is planned for DataHUB. Various projects within the NFDI and Science Data Centers in Baden-Württemberg are using InvenioRDM, which will result in collaborations. This will also ensure the availability of personnel with the appropriate expertise over the long term.

The deployment of InvenioRDM requires close coordination and cooperation between the players involved at all levels. This includes contact with the developer community as well as coordination with the storage infrastructure operators of DataPLANT, and the institutions that provide the organizational framework for user authentication and the DOI interface.

3.F. ARC METADATA REGISTRY

The registry is a tool for integrated search and analysis of individual ARCs and experimental metadata. The web-based user interface provides a consolidated real-time view of the public ARCs within the DataPLANT community. The search functions of the ARC Registry application enable users to explore the ARCs and the assay/measurement data. The users can look up the ARCs by specific keywords, such as the working group they are interested in or by a particular data steward from the group. Most importantly, the users can also search and explore the experimental metadata from various ARCs simultaneously, thus paving the way toward cross-omics metadata exploration within the community.

The registry application receives data through push messages upon ARC updates from individual GitLab instances. It provides the latest snapshot of the ARCs across all DataHUBs at a given time. Also, it presents the evolution of the individual ARCs by keeping track of the history of the ARC updates.

4. WORKFLOW INTEGRATION

As the central data management component, the DataHUB is set to act as a starting point for the various analysis workflows created by the scientists of the various disciplines. The workflow description is stored in the ARC to be made accessible to processing frameworks like Galaxy[ix] or nf-core nextflow pipelines[x].

The ARCfs component is one possibility to facilitate such access. This is achieved by providing a file system-like view on ARCs to Galaxy and possibly other services in the future. More concrete, it is a read-only file system abstraction for Python using PyFilesystem2[5]. It functions by providing file system typical methods to a developer or framework. Contrary to some file system views on Git, it does not use local repositories[xi] or lazily makes files available locally[xii], but exclusively communicates with GitLab through its REST API to retrieve file-metadata or -content, including LFS files.

Since the Galaxy platform already supports various PyFilesystem2 file systems, developing one which allows to access ARCs, or GitLab repositories in general, seems promising. This approach has several advantages as it enables users to browse and load content of all ARCs they have access to, through a simple web interface. This avoids the need to search for specific ARC repositories using the GitLab website or other tools. But more importantly, the need to download files locally, just to upload them again, is eliminated. Instead, files are transferred directly from the GitLab- to the Galaxy server. Furthermore, manually downloading and uploading files is not only time-consuming, but could potentially involve partial clones of repositories, which is comparably complex. Such partial clones may be necessary though, since ARCs can become large and, potentially, only a subset of files is needed.

ARCfs is currently in a prototyping stage and is being tested as a workflow integration in Galaxy. To access public ARCs hosted on DataHUB, there is

[5] Project website: https://github.com/PyFilesystem/pyfilesystem2

no need for the user to configure anything. To gain access to private repositories, a GitLab access token with the scope read api must be provided. To enable users to write computation results from Galaxy back into an ARC directly, there is currently a version of ARCfs with read-write access in development. Right now, it is only possible to inspect the main branches of the ARC repositories. Including an option to view specific tags or releases may be an improvement for future versions.

4.A. STABLE BACKEND STORAGE INFRASTRUCTURE

The publication of research data requires its security and permanent accessibility. The use of InvenioRDM is therefore based on the bwSFS[xiii] Storage-for-Science system for scientific data, which can store data geo-redundantly and long-term[6] bwSFS was acquired for the efficient and long-term secure storage of research data, in addition to the already existing repositories of the various scientific disciplines. With bwSFS, the infrastructural resources for research data management are bundled in order to better support the implementation of specific FDM requirements. The system is federated across the sites of the participating university computer centers, the core infrastructure providers in DataPLANT and BioDATEN[7]. bwSFS has a solid, expandable hardware base with advanced monitoring and various redundancies, some of which extend beyond site boundaries, in the form of full mirroring of the file system area and erasure coding for object storage.

bwSFS provides nearly 20 petabytes of usable storage in the form of network file systems and object storage. The system works with capacity optimization through compression and deduplication, so that additional virtual capacity is available, especially for unpacked data, which is still frequently used in biology. The file systems are primarily provided locally at the main sites or, via a caching component, also transparently locally for workgroups via cache outposts in Stuttgart and Konstanz. The DataHUB uses both NFS for GitLab caching and object storage for LFS objects and InvenioRDM repository operation. In order to provide both a stable and efficient storage infrastructure for large amounts of data, a cross-site S3 erasure encoding is configured. Parts of the object repository will be available worldwide, especially for use in distributed workflows and collaborations.

Beside the centralized storage, the DataHUB is designed to be deployable on-premise as well to integrate local data storage alternatively. This caters to the requirements when handling sensitive data, which should not leave the premises of the research institution.

Another crucial criterion is the backup of both the users' research data and the various application data of the DataPLANT services. Both types of data need to be handled differently. The users' research data is stored in a geo-redundant bwSFS S3 region. While a common backup strategy for the application data of services (i.e. using database dumps) is appealing, it would require a shutdown of the services to assure a consistent state. Therefore, the application/ service specific backup tools should be used whenever possible. These tools usually guarantee a consistent backup of the application state, even if the backups are created while the service is running. The backups themselves are also stored in a mirrored bwSFS S3 bucket, which has higher redundancy than the erasure-coding buckets for the users' research data (which would be too large to mirror completely).

5. CONCLUSION AND OUTLOOK

The DataPLANT DataHUB is running productively since the end of 2021, hosts 159 users and tracks 216 ARCs with around 8.8 TB of total data. GitLab is a powerful framework which offers a wealth of features and user interfaces to be useful outside the core software development community. As an open source project, it can be modified to the needs of research data management to a certain degree. Nevertheless, there are features that are only available in the premium version of GitLab that ensure automatic merging of requests with automated code quality checks. The same applies in general to automatic merge approval rules. The process of circumventing the constraints with custom tools and specially developed procedures could become very time-consuming. Also, it

[6] bwSFS is a distributed system maintained by the Universities of Freiburg and Tübingen, and, in the near future, also Stuttgart and Hohenheim.

[7] One of the founding partners in DataPLANT, see https://portal.biodaten.info

is not clear how often code changes from the open source GitLab version will need to be patched and how long these code changes will remain compatible with the original.

Some of those features can be replaced with own services. One reason to develop a microservice for the ARC publication workflow is the missing feature of external credentials in the open source version of GitLab. The possibility to use secure API tokens in a GitLab pipeline only exists in the Premium version, with which the software Vault from Hashicorp can be used. Without that feature, users could extract credentials from the CI pipeline. Additionally, the publication workflow developed within DataPLANT using GitLab's CI/CD pipelines and the publication badge might not be obvious to all users. A more user-friendly way could be to create an OAuth application in GitLab for the publication microservice. The application would then get an access token for the logged-in user and could present an overview of the ARCs that available for publication. Alternatively, an InvenioRDM module to integrate the available ARCs from GitLab in a publication form on the InvenioRDM platform (akin to Zenodo's) could be envisioned but would require additional development work to be realized. This workflow is continuously evaluated during operation.

In general, GitLab as a science gateway would be a valuable addition to the NFDI software landscape. Thus, it would be beneficial to address it as a joint service in cross-domain activities of all interested consortia.

6. ACKNOWLEDGMENT

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.

7. REFERENCES

i C. Garth, J. Lukasczyk, T. Mühlhaus, B. Venn, J. Krüger, K. Glogowski, C. Martins Rodrigues, and 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 and N. Bisheh, Eds. Heidelberg, Germany: heiBOOKS, 2022, pp. 366–373. https://doi.org/10.11588/heibooks.979.c13751

ii C. Martins Rodrigues, D. von Suchodoletz, T. Mühlhaus, J. Krüger, and B. Usadel, »DataPLANT—Ein NFDI-Konsortium der Pflanzen-Grundlagenforschung,« *Bausteine Forschungsdatenmanagement,* vol. 2, pp. 46–56, 2021. https://doi.org/10.17192/bfdm.2021.2.8335

iii 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, no. 2, pp. 29, 2016. https://doi.org/10.21105/joss.00029

iv N. Hartl, E. Wössner, and Y. Sure-Vetter, »Nationale Forschungsdateninfrastruktur (NFDI),« Informatik Spektrum, vol. 44, p. 370–373, 2021. [Online]. Available: https://doi.org/10.1007/s00287-021-01392-6

v B. Venn, K. Schneider, K. Frey, H. L. Weil, J. Werner, F. Wannenmacher, T. Zajac, D. von Suchodoletz, B. Usadel, J. Krüger, C. Garth, and T. Mühlhaus, »Fostering the democratization of research data by using the Annotated Research Context (ARC) as practical implementation,« presented at E-Science-Tage 2021: Share Your Research Data, 2021, doi: https://dx.doi.org/10.11588/heidok.00029769

vi C. Garth et al., *E-Science-Tage 2021: Share Your Research Data*, 2022, pp. 366–373.

vii T. Mühlhaus, D. Brilhaus, M. Tschöpe, O. Maus, B. Grüning, C. Garth, C. Martins Rodrigues, and D. von Suchodoletz, »DataPLANT—Tools and Services to structure the Data Jungle for fundamental plant researchers,« in *E-Science-Tage 2021: Share Your Research Data*, V. Heuveline and N. Bisheh, Eds. Heidelberg, Germany: heiBOOKS, 2022, pp. 132–145. https://doi.org/10.11588/heibooks.979.c13724

viii DataPLANT Consortium Contributors, »Sample Annotated Research Context. This is a minimal Example ARC packaging an mRNA-Seq dataset with metadata and computations.« Accessed: [Month, day] 2023. [Online]. Available: https://git.nfdi4plants.org/brilator/samplearc rnaseq

ix Galaxy Community, »The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update,« *Nucleic Acids Research*, vol. 50, no. W1, pp. W345–W351, 2022. https://doi.org/10.1093/nar/gkac247

x P. Ewels, A. Peltzer, S. Fillinger, H. Patel, J. Alneberg, A. Wilm, M. Garcia, P. Di Tommaso, and S. Nahnsen, »The nf-core framework for community-curated bioinformatics pipelines,« *Nature Biotechnology*, vol. 38, pp. 276–278, 2020. https://doi.org/10.1038/s41587-020-0439-x

xi V. Salis and D. Spinellis, »RepoFS: File system view of Git repositories,« *SoftwareX*, vol. 9, pp. 288–292, 2019. https://doi.org/10.1016/j.softx.2019.03.007

xii J. Schroeder, »GitOD: An on demand distributed file system approach to Version Control,« in *2012 International Conference on Collaboration Technologies and Systems (CTS), IEEE*, 2012, pp. 613–615. https://doi.org/10.1109/CTS.2012.6261115

xiii D. von Suchodoletz, U. Hahn, J. Bauer, K. Glogowski, and M. Seifert, »Storage for Science—Aktueller Stand und anstehende Entwicklungen eines verteilten FDM-Systems,« in *E-Science-Tage 2021: Share Your Research Data,* V. Heuveline and N. Bisheh, Eds. Heidelberg, Germany: heiBOOKS, 2022, pp. 298–305. https://doi.org/10.11588/heibooks.979.c13741

Proceedings of the 15th International Workshop on Science Gateways

The conferences hosted by the International Workshop on Science Gateways (IWSG) have a long-standing tradition. The workshop series aims to advance in the field of science gateways and to improve and make services more accessible to researchers in various fields. The IWSG 2023 included six full-paper presentations and was complemented by nine lightning talks. These contributions spanned multiple fields, from biology to astronomy and beyond, showcasing a variety of tools and methodologies.

9 783989 450196

TÜBINGEN
UNIVERSITY
PRESS