SLICES Data Management Infrastructure for Reproducible Experimental Research on Digital Technologies

Yuri Demchenko
University of Amsterdam
The Netherlands
y.demchenko@uva.nl

Sebastian Gallenmüller
Technical University of Munich
Germany
gallenmu@net.in.tum.de

Serge Fdida
Sorbonne University
France
serge.fdida@sorbonne-universite.fr

Thijs Rausch
University of Amsterdam
The Netherlands
thijs.rausch@student.uva.nl

Panayiotis Andreou
UCLan Cyprus
Cyprus
PGAndreou@uclan.ac.uk

Damien Saucez
Inria
France
damien.saucez@inria.fr

Abstract—**This paper presents the ongoing research effort related to the design of the Data Management Infrastructure (DMI) to support experimental research on digital technologies with application to the ESFRI SLICES scientific instrument. We consider the experiment documentation and data collection across the whole continuum of access network, IoT, edge, cloud, and data processing workflow. The paper includes the requirements analysis for DMI to enable research reproducibility of complex and large-scale experimentation. We provide an analysis of data collected and processed in SLICES and explain approaches and solutions used in SLICES for experimental research reproducibility, primarily based on the plain orchestration service and supported by metadata collection tools. The proposed multi-layer DMI includes: data (storage) access, data processing, data ingest, experiment management, and virtual research environment. The paper also provides recommendations for the selection of existing standards and tools for data and metadata management, in particular those developed by EOSC and supported by the RDA community to ensure wide compatibility and integration.**

Keywords—Experimental Research Reproducibility, Experimental Data Management Infrastructure, FAIR data principles, Metadata management, SLICES Research Infrastructure.

I. INTRODUCTION

Wider adoption of Open Science requires a modern research infrastructure and scientists to pay more attention to consistent data management in order to support effective data sharing and communication between researchers [1]. Introducing FAIR data principles and ongoing development and implementation of supporting standards, frameworks, and tools in recent years, significantly improved the possibility for sharing research data and research results, targeting research reproducibility, sharing data, or other publishable research results via the popular Open Access or self-archiving services OpenAIRE [2] and Zenodo [3]. The European Open Science Cloud (EOSC) [4] provides the federated data sharing infrastructure. Recent developments such as RO Crate [5, 6] have the potential of supporting complex research objects and their evolution. This is especially important for experimental research reproducibility that requires documenting a large volume of information related to the experiment setup, workflow, input data, and measurement data [7].

SLICES Research Infrastructure (SLICES-RI) [8] is dedicated to experimental research on new digital infrastructure technologies that power modern data driven science, which fast development requires continuous experimentation to acquire practical knowledge, gain experience and develop design patterns. This paper is focused on the definition and design approach of the SLICES Data Management Infrastructure (DMI) to support experimental studies on digital infrastructure technologies.

The paper is structured as follows. Section II provides information about EOSC and important projects supporting new advanced features for FAIR data sharing and metadata management. Section III introduces the SLICES-RI and describes data management aspects for experimental research reproducibility. Section IV describes the experimental data lifecycle stages, key requirements of the DMI, and the proposed DMI architecture. Section V describes the metadata required for the SLICES infrastructure and experiment description to ensure experiment reproducibility. Section VI provides information about the ongoing implementation of the experiment reproducibility with the plain orchestration service (pos). Section VII discusses which EOSC tools and services can be used in SLICES to support metadata management and research reproducibility. The paper concludes with a summary and recommendations for future research in Section VIII.

II. EOSC AND FAIR DATA MANAGEMENT TOOLS

EOSC is an initiative and programme by the European Union to provide European researchers, innovators, and citizens with a federated and open multi-disciplinary environment where they can publish, find and re-use data, tools and services for research, innovation, and educational purposes. The EOSC Strategic Research and Innovation Agenda provided a roadmap to achieve the EOSC vision and objectives, namely to deliver an operational "Web of FAIR data and services" for science [9]. The EOSC Portal includes the Marketplace and Catalog of services and resources that are offered by European research institutions which can be used by other projects and researchers [10].

New EOSC projects contribute to creating a foundation for research data interoperability and implementation of the FAIR data principles. In particular, the FAIRCORE4EOSC project

[11] develops a set of important tools for consistent FAIR implementation and data and metadata management, of which the EOSC Metadata Schema and Crosswalk Registry (MSCR) addresses publication, discovery, and access of metadata schemas and provide functions to operationalize metadata conversions.

The RELIANCE project [12] extends the Research Object (RO) with FAIR compliance for the Research Lifecycle Management. ROHub is a service by RELIANCE for the storage, lifecycle management, and preservation of scientific research, campaigns via research objects [13].

RO-Crate provides a framework for packaging research products into FAIR Research Objects to make them discoverable, executable, reproducible [6, 14]. RO-Crate provides a metadata schema and a packaging structure for typical research products such as workflows, software, models, presentations, articles, and data, which makes RO-and RO-Crate directly applicable for SLICES purposes [8].

III. SLICES-RI AND EXPERIMENTAL DATA MANAGEMENT

A. *SLICES Research Infrastructure*

The Scientific Large-scale Infrastructure for Computing/ Communication Experimental Studies (SLICES) [8] is a distributed Digital Infrastructure designed to support large-scale experimental research focused on networking protocols, radio technologies, services, data collection, parallel and distributed computing and, in particular, cloud and edge-based computing architectures and services. This encompasses the full range of network, computing, and storage functions required for on-demand services across many verticals and addresses new complex research challenges, supporting disruptive science in IoT, networks and distributed systems. SLICES will integrate multiple experimental facilities and testbeds operated by partners, providing a common services access and integration platform. SLICES will allow academics and industry to experiment and test the spectrum of digital technologies whereby the computing, network, storage, and IoT resources can be combined to design, experiment, operate, and automate the full research lifecycle.

B. *Experiment Reproducibility as a Service in SLICES*

SLICES will support experimental research reproducibility as one of the core principles of Open Science. The primary focus will be on the **repeatability** and **reproducibility** with the future support of **replicability** and shared distributed experiments orchestration [7]. Reproducibility of experimental research imposes additional requirements on the reproducible experiment setup, including resource provisioning, experiment environment setup, and experiment and data lifecycle management. The following aspects will be addressed:

- Documenting all relevant parameters and environment for experiments,
- Automate the documentation of experiments; a well-structured experiment workflow may serve as documentation.

Offering Experimental Research Reproducibility as a Service (ERRaaS) will be beneficial to the research community by:

- Reducing the amount of work for experimenters to create reproducible experiments,
- Diminishing the load for other researchers to recreate and re-run experiments,
- Decreasing the overall energy consumption and environmental impact of large-scale and complex experiments
- Automating the entire experiment (setup, execution, evaluation), including energy optimization.

Making reproducibility an integral part of the experiment design will serve another purpose of documenting infrastructure design and usage patterns that can be re-used by other RIs intending to use new DI technologies in their research.

C. *Support of FAIR Data principles in SLICES*

FAIR data management principles [15] are realized primarily via consistent metadata definition and management for research data constituting the research outcomes. FAIR principles are currently extended for all research products, digital objects, and software [16].

FAIR implementation requires complex infrastructure services to support data storage, search, access, and processing. The following lists the required infrastructure services to support corresponding FAIR principles:

Findability is supported via metadata publication and discovery:

- Metadata registries and discovery infrastructure
- PID (Persistent IDentifier) and handles infrastructure

Accessibility requires a complex set of services:

- Repositories and data storage, supporting open data access protocols and APIs
- Access and usage policies, supporting data sovereignty
- Data protection, compliance, privacy, and GDPR

Interoperability is ensured by well-defined and open:

- Standard data formats and metadata schema registration
- FAIR maturity levels and certification

Reusability requires full documentation of the data origin and research reproducibility:

- Metadata, PID, and API linked or embedded into datasets
- Research workflow and/or experiment description
- Data provenance and lineage

The following sections explain how these requirements are implemented in the SLICES DMI.

IV. SLICES DATA MANAGEMENT INFRASTRUCTURE

A. *Experimental Data Management stages*

Management of experimental data is a key aspect of SLICES-RI, and it includes several services that must support all stages of the experimental data lifecycle. As illustrated in Figure 1, SLICES-RI will operate distributed federated Data Storage and Management Infrastructure to support activities typical for experimental research, such as experiment planning and deployment (as explained in the previous sections), the discovery of data from internal data archives and external sources and data publication.

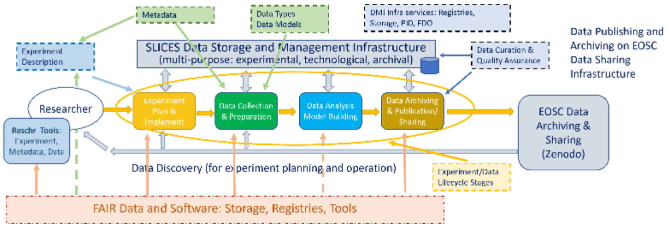


Figure 1– SLICES Data Management stages and supporting infrastructure.

SLICES DMI establishes policy for data governance and management, including data security and quality assurance (data curation), that are supported by corresponding infrastructure tools. Figure 1 illustrates stages and activities where the FAIR-compliant metadata must be applied.

Each data lifecycle stage, i.e., experiment setup, data collection, data analysis, and finally, data archiving, typically works with its own datasets, which are linked and their transformation must be recorded in the process that is called lineage (which can also be extended to provenance for complex linked scientific data). All staged datasets need to be stored and possibly re-used in later processes.

Many experiments may require already existing datasets that will be available in the SLICES data repositories or can be obtained/discovered in EOSC data repositories as illustrated by links to EOSC data services.

B. Requirements to support the experimental data management

Data Management is an essential component of the SLICES-RI infrastructure that includes data collection from experiments (including experiment description and measurement data), data storage, data preparation, data lineage and quality assurance, data publication, and data sharing.

The following are requirements for SLICES DMI for experimental data issued from best practices and use cases analysis in the SLICES-DS project [8, 25]:

RDM1. Distributed data storage and experimental data(set) repositories should support common data and metadata interoperability standards, in particular, common data and metadata formats. Outsourcing of data storage to the cloud must be protected with appropriate access control and compliant with the SLICES Data Management policies.

RDM2. SLICES DMI should support the whole research data lifecycle. It should provide interfaces to experiment workflow and staging.

RDM3. SLICES DMI shall provide PID (Persistent IDentifier) and FDO (FAIR Digital Object) registration and resolution services to support linked data and data discovery that should be integrated with EOSC services.

RDM4. SLICES DMI must support (trusted) data exchange and transfer protocols that allow policy-based access control to comply with the data protection regulations.

RDM5. SLICES DMI must enforce user and application access control and identity management policies adopted by the SLICES community that can be potentially federated with the EOSC Federated AAI.

RDM6. Procedures and policies must be implemented for data curation and quality assurance.

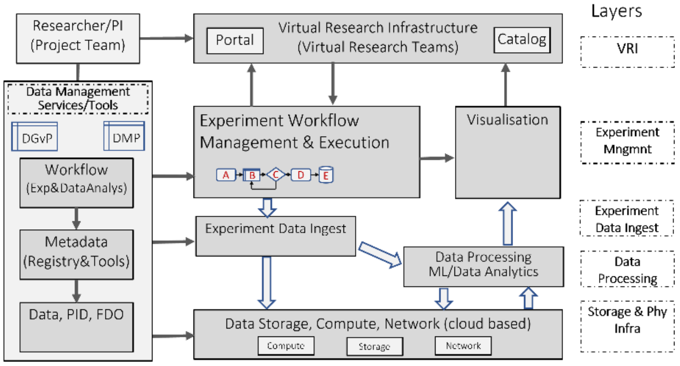
RDM7. Certification of data and metadata repositories should be considered at some maturity level following certification and maturity recommendations by RDA.

SLICES DMI will be designed in such a way that would allow integration with the EOSC federated data infrastructure and services to allow a hybrid data management infrastructure that may include both its own data storage, as part of the private cloud, and external data storage offered by EOSC and EGI [17] communities. The use of public cloud storage and file sharing services will be regulated by data management policies.

C. SLICES DMI Architecture

The consistent definition of DMI will impose specific requirements to the SLICES Reference Architecture and will require the implementation of special services to support data collection, data management, and data sharing at all functional layers of the SLICES infrastructure.

DMI creation will have a staged process starting with the bottom-up data and metadata services integration with the existing SLICES testbeds and experimental sites, delivering a Minimum Viable Product (MVP). DMI will follow the SLICES-RI evolution and incorporate new data and metadata tools development, primarily coordinated and facilitated by EOSC. The long-term vision for DMI should incorporate all these factors by adopting a sustainable architecture design principles.



DGvP – Data Governance Policy DMP – Data Management Plan

Figure 2. Architecture of the SLICES Data Management Infrastructure

DMI Architecture definition includes hierarchical service layers (allowing horizontal and vertical composition and integration) and cross-layer services defined as planes. Such architecture definition allows separating data management and governance functions, concerns, and actors/roles. The following layers and planes are defined:

Layer 5 – Virtual Research Environment (VRE) and researcher portal or dashboard.

Layer 4 - Experiment configuration and management.

Layer 3 - Experimental data collection/recording that applies data models and metadata for experimental data.

Layer 2 - Data processing that performs data analysis, allows ML models building for processes and systems-under-test, and ensures the computation workflow scalability and portability.

Layer 1 - Data Storage, Archiving, Exchange that represents the physical or virtual infrastructure resources for data or metadata storage, archiving and publication. This layer supports FAIR Digital Object (FDO), PID registries and gateway/proxy.

The Data Management Plane includes Data Management Services and Tools that can be used by each of the DMI layers:

- Data Management Plan and Data Quality Assurance, FAIR compliance,
- Metadata registries and tools,
- Data Governance Policy, Data Security, GDPR compliance.

VRE and user portal may benefit from implementing the Platform RI as a Service (PRIaaS) architecture that is compliant with the TeleManagement Forum Digital Platform Reference Architecture for telecom system [18].

V. METADATA TO DESCRIBE INFRASTRUCTURE AND EXPERIMENTS

A. General Metadata Definition and Services

Metadata are an important component of DMI that provide a basis for services interoperability, experimental research reproducibility, effective data sharing and discovery. Effective and consistent metadata management is the foundation of the FAIR data principles implementation. All data are defined by the data models, metadata, data formats and data types. Metadata are defined as part of the data model.

For SLICES as a RI for experimental studies in digital technologies and ICT, metadata includes three main areas:

- General services description: metadata profiles and metadata will be used for publishing SLICES services in EOSC Catalog and the SLICES services catalog.
- Description of data collected, produced and handled in SLICES-RI that include experimental data, staged/processed data, archival data, publications, reports, and management data. Additional data categorization is required.
- Experiment description that includes all necessary information for experiment reproducibility and deployment.

Two other categories of metadata may be required to support SLICES experiments include:

- Infrastructure descriptions that are required for infrastructure management and monitoring (network devices, network traffic, status and events). This type of metadata is well supported by existing network and service management standards (SNMP MIB-II, DMTF CIM and CIMI)
- Metadata for data processing and lineage, in particular, for data used in ML and AI processes.

Defining domain-specific metadata requires the definition of the metadata schema and namespaces that create a basis for unique metadata elements identification and consequently discovery, sharing and integration.

B. Experiment Description and Metadata

The experiment description and corresponding metadata must ensure experiment reproducibility and FAIR-compliant experimental data sharing.

The following data types and metadata are considered as essential for consistent experiment description:

- Experiment abstract model with parameters, input variables and variables under test (defined at the beginning),
- Experiment setup/infrastructure, including network equipment and the network topology, including VMs/containers, that should cover hardware, firmware, and software,
- Configuration of all infrastructure components, deployment sequence (presumably in the form of Ansible playbooks, Terraform plans, or Jupyter Notebooks),
- Test generators, measurement equipment and sensors (and corresponding infrastructure points), including specification of the generated traffic and its patterns,
- Experiment workflow (the usage of pos ensures reproducibility of experiment workflow),
- Data ingest process, data preprocessing and assessment,
- APIs for experiment setup, monitoring, and data collection.
- Data models and metadata must be defined for all types of data describing the experiment.

C. SLICES Metadata Profile and Metadata Registry Service

The SLICES general metadata profile has been defined in the SLICES-DS project (refer to SLICES-DS Deliverables D4.3 [19] and D4.5 [20]) and includes different element categories: (i) general, (ii) type-specific, and (iii) domain-specific, to meet FAIR compliance, services interoperability, and experimental research reproducibility. It defines a metadata structure for documentation, access and reuse of any digital object managed by DMI, such as data, services, and experiments.

The *general category* includes human- and machine-readable domain-agnostic elements to facilitate discovery (e.g., persistent identifier, name, creator) and access of any type of digital object. Discovery is enabled through the Metadata Registry System (MRS) catalogue, where all digital objects are registered. This enables SLICES to publish any compatible object to other catalogues, such as EOSC, using appropriate metadata adaptors.

The type-specific category includes metadata describing entities present in and constituting the SLICES infrastructure. This is necessary to describe complex experimental setups, which may involve multiple, heterogeneous, and geo-diverse equipment distributed over multiple sites and testbeds. The domain-specific category includes the metadata describing entities from the type-specific category.

VI. EXPERIMENT AUTOMATION AND REPRODUCIBILITY

A. Tools for Experiment Control and Lifecycle Management

In this section, we refer to popular experiment control and workflow management frameworks and tools. A more detailed review is provided in the authors' paper [7].

Multiple projects built and operated frameworks for research on networking, IoT, and distributed computing, such as Fed4FIRE [21], OneLab (EU) [22], Planetlab (global) [23], or GENI [24] mainly focusing on resource allocation for experiments. There are also more high-level approaches to defining and execute portable experiment workflows, such as OMF [25] or NEPI [26].

GitHub's functionality for version control and sharing code [27] can be used for managing scientific code and data, and for running experiments and processing experimental data. GitHub tools are often combined with the infrastructure deployment using Ansible playbooks [28] or Terraform plans [29].

Jupyter Notebooks [30] are widely used for data analysis and reporting. Recent developments target the full scientific research cycle, including experiment development and exploration. Jupyter Notebooks were used in the Fed4FIRE+ project by several testbeds to offer reproducible experiments.

The Chameleon cloud platform [31] is a large-scale experimental platform supporting experimental workflows for computer science research. The Chameleon experimental workflow uses Jupyter Notebooks and includes stages related to resource discovery, allocation, dynamic configuration, orchestration, and monitoring. The library of orchestration templates and images is available on the Chameleon website [32].

Higher level experiment workflow definition and cross-platform portability can be achieved using the popular Common Workflow Language (CWL) [33]. CWL describes computational workflows and needs a so-called runner to execute them. A reference implementation of such a runner is cwltool [34], however, several workflow management systems implement CWL support, e.g., Apache Airflow [37] or StreamFlow [36]. The Galaxy workflow management system [36] is popular among the research community and maintains a repository of scientific workflows.

B. *Plain orchestrating service (pos)*

SLICES will use the pos controller [38, 39] to provide a basis for developing an experiment management platform. Therefore, we plan to extend pos to support an experiment workflow definition and experimental data documentation in a FAIR compliant way.

A key feature of pos is the creation of reproducible experiments. Pos ensures that experiment nodes always boot into the same, well-defined state. Additionally, testbed users are required to automate the entire experiment workflow. Both the automation and the well-defined starting point are enforced by the pos framework, ensuring the creation of reproducible experiments. Reproducibility is a core part of the experiment design; hence, we call this property reproducibility by design.

Pos consists of two components: (1) a *testbed controller* and (2) a framework to express *experiment workflows*. The testbed controller provides basic functions to manage resources, such as user authentication, reservation of resources, and configuration of the allocated resources. The experiment workflow manages the actual execution of experiments including steps such as the experiment-specific configuration of experiment nodes, the synchronization of the workflow between different experiment nodes, or the execution of the actual measurements. Other frameworks, focus on one of the two components, e.g., GENI [24] or Chameleon [31] that provide the services of testbed controller, or OMF [25] and NEPI [26] that execute experiment workflows.

Coordinated development of both components, like it is done with pos, offers significant benefits from a data management perspective compared to two separately developed components. As a testbed controller, pos has access to information such as the used hardware or the network topology collected via standard Linux tools such as `lshw` or `lldpd`. As a framework defining the experiment workflow, pos gathers additional data on the execution of the experiment, such as the installed software or the parameters investigated during a specific measurement. In pos, data collection can be more comprehensive and fully automated. As an example, pos uses JSON a widely used data format to provide the collected information about the hardware, as reported by the lshw tool.

C. *Types of f Data in POS Experiment*

A typical example of a pos experiment can be found on GitHub [46]. It contains various files describing an experiment:

- Scripts (defining the workflow of the experiment)
- Configuration variables (`global` and `local` variables)
- Measurement variables (`loop` variables)
- Measurement data (results)
- Environment (experiment metadata, hardware config)

Pos experiments follow a well-defined experiment workflow separated into three phases (setup, task description and evaluation). In the first phase, the experiment script initiates and prepares the experiment execution. It is executed on the management host. Separate scripts configure the participating experiment nodes using global and node-specific (local) variables. The second phase consists of repeated execution of the task description script on all participating experiment nodes. Each run uses a specific set of loop variables to parameterize the executed measurements. All the measurement data is collected in a folder created for a specific experiment workflow execution including measurement data, used hardware or the start of the experiments. The third phase evaluates the measured data from the result folder, typically using a plotting script additionally taking into account the loop variables used to create a specific measurement. The data of the experiment workflow, i.e., the scripts, variables, data and environment, can be used to create a bundle to be released for publication. This step can be automated with the structure we use for the experiment workflow.

VII. Researcher Tools for Metadata Management

A. *Extending RO Crate for Experiment Description*

SLICES will adopt the RO [19] and RO-Crate [5. 6, 14] frameworks for packaging and managing experimental research products and documenting their evolution (provenance) to benefit from the rich expertise of the RO-Crate Community.

RO-Crate Specification Version 1.1 [6] allows packaging the following information/entities: metadata, workflows, software, models, data, publications, presentations, metadata, logs **-** all entities can be local or linked. A resource is stored using RO-Crate directory with the following structure:

```
<RO-Crate root-directory>/
  |  ro-crate-metadata.json # Metadata file MUST
  |  ro-crate-preview.html  # RO-Crate website MAY
  |  ro-crate-preview_files/  # MAY be present
  |    | [other RO-Crate website files])
  |  [payload files and directories]  # 0 or more
```

The metadata file uses the JSON format for Linked Data (JSON-LD) [40] and provides information about all entities included in the RO-Crate. Although context attributes of the data entity can be used to document equipment or software used to create files, the description is limited to textual description and serial numbers. Provenance information is limited to CreateAction and UpdateAction attributes of the data entity.

SLICES will follow the formal RO-Crate procedure to create a new ROE-Crate profile/schema for Experimental RO that, in particular, would support all necessary information required for the full experiment description and reproducibility.

B. *Tools and Platforms for Infrastructure and Experiment Metadata Management*

SLICES will use Open Source Metadata Registries and metadata management tools to implement the initial metadata profiles for services and project outcomes defined by the SLICES-DS project. So far, we reviewed Apache Atlas [41], Amundsen [42], and OpenMetadata [43]. We will investigate the possibility and benefits of using the EOSC Metadata Schema and Crosswalk Registry (MSCR) [44] for managing infrastructure and experiment related metadata. All tools allow new metadata schemas in compliance with the schema.org and formats JSON/JASON-LD, RDF, Microdata, etc. It is important that metadata tools can interoperate with RO/ROE Crate tools.

SLICES Experiment metadata tools should allow the extraction of metadata from the experiment description, including deployment scripts (for example, shell scripts, Ansible playbooks, Terraform plans), workflow, processing and analytics. The majority of existing metadata extraction tools either extract built-in metadata, or search for metadata files in the code directory. For the particular case of experiment reproducibility, we need to extract metadata related to variables and operations in the experimental data processing applications.

The Software Heritage Deposit service and application [45], which is a product from the FAIRCOR4ESOC tools set, provides a good initial basis for extracting metadata from research software both generic and software specific [46], but needs to be extended with the functionality mentioned above.

VIII. Conclusion and Further Developments

This paper presents ongoing research and developments carried out in SLICES-RI regarding the Data Management Infrastructure for experimental research data and experiments reproducibility. The paper presents an analysis of the data produced in SLICES experimental facilities and testbeds, analyses experimental data lifecycles and proposes a multilayer DMI architecture that includes data storage and access, data processing, experimental data collection, experiment management, and VRE.

The important contribution of the paper is the detailed analysis of data required for experiment description, configuration, and execution. We also present the plain orchestration services that provides a platform for SLICES experiment automations and Experimental Research Reproducibility as a Service.

Finally, we provide an analysis of the existing Open Source tools to support consistent data and metadata management in order to support experimental research reproducibility and FAIR and Open Science compliance. The majority of selected tools are from the EOSC toolset that should ease the integration with the EOSC data management and sharing ecosystem.

Acknowledgment

The research leading to these results has received funding from the Horizon Europe projects SLICES-DS (951850), SLICES-SC (101008468), and SLICES-PP (951850).

References

[1] Open Science [online] https://www.fosteropenscience.eu/content/what-open-science-introduction

[2] OpenAIRE [online] https://www.openaire.eu/en/home

[3] Zenodo. [online] https://zenodo.org/

[4] EOSC Association [online] https://eosc.eu/about-eosc

[5] Research Object Crate (RO-Crate) [online] https://www.researchobject.org/ro-crate/

[6] RO-Crate Metadata Specification 1.1 [online] https://www.researchobject.org/ro-crate/1.1/

[7] Demchenko, Y., S. Gallenmüller, S. Fdida, P. Andreou, C. Cretaz, M. Kirkeng, Experimental Research Reproducibility and Experiment Workflow Management. TASIR Workshop, Proc. COMSNETS 2023 Conf.

[8] S. Fdida, N. Makris, T. Korakis, R. Bruno, A. Passarella, P. Andreou, B. Belter, C. Crettaz, W. Dabbous, Y. Demchenko, and R. Knopp, "Slices, a scientific instrument for the networking community," *Comput. Commun.*, vol. 193, pp. 189–203, 2022.

[9] Strategic Research and Innovation Agenda (SRIA) of the European Open Science Cloud (EOSC), Version 1.0, 21 June 2021 [online] https://op.europa.eu/nl/publication-detail/-/publication/f9b12d1d-74ea-11ec-9136-01aa75ed71a1

[10] EOSC Portal Catalog and Marketplace [online] https://marketplace.eosc-portal.eu/

[11] EOSC Components, FAIRCORE4EOSC Project [online] https://faircore4eosc.eu/eosc-core-components

[12] RELIANCE Project [online] https://www.reliance-project.eu/

[13] ROHub [online] http://reliance.rohub.org

[14] Soiland-Reyes, S., et al, Packaging research artefacts with ROCrate. Data Science, 5(2), 97-138. https://doi.org/10.3233/DS-210053

[15] FAIR Data Principles [online] https://www.go-fair.org/fair-principles/

[16] RDA FAIR Data Maturity Model Working Group (2020). FAIR Data Maturity Model: specification and guidelines. Research Data Alliance. DOI: 10.15497/RDA00050

[17] EGI [online] https://www.egi.eu/

[18] Y. Demchenko, C. de Laat, W. Los, and L. Gommans, "Defining platform research infrastructure as a service (PRIaaS) for future scientific data infrastructure," in Designing Data Spaces: The Ecosystem Approach to Competitive Advantage, pp. 241–260, Springer, 2022.

[19] Deliverable D4.3 Definition of the SLICES metadata profiles to support FAIR principles (initial proposal), SLICES-DS Project. 31 August 2021.

[20] Deliverable D4.5 SLICES infrastructure and services integration with EOSC, Open Science and FAIR: Recommendations and design patterns (final report)/. SLICES-DS Project. 31 August 2022.

[21] Fed4FIRE [online] https://www.fed4fire.eu/.

[22] OneLab [online] https://onelab.eu/.

[23] Planetlab [online] https://www.planet.com/. Last accessed: 2022-11-16.

[24] Geni [online] https://portal.geni.net/. Last accessed: 2022-11-16.

[25] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar, "OMF: a control and management framework for networking testbeds," *ACM SIGOPS Oper. Syst. Rev.*, vol. 43, no. 4, pp. 54–59, 2009.

[26] A. Quereilhac, M. Lacage, C. D. Freire, T. Turletti, and W. Dabbous, "NEPI: an integration framework for network experimentation," in *19th Int Conf on Software, Telecommunications and Computer Networks, SoftCOM 2011, Split, Croatia, September 15-17, 2011*, pp. 1–5.

[27] GitHub Features [online] https://github.com/features

[28] Redhat Ansible [online] https://www.ansible.com/

[29] Terraform [online] https://www.terraform.io/

[30] Jupyter [online] https://jupyter.org/

[31] Chameleon Cloud: Experiment in the Edge to Cloud continuum https://www.chameleoncloud.org/media/filer_public/8d/a8/8da8b517-fd99-46ce-94ea-61d4edd94531/cluster.pdf,

[32] K. Keahey, et al, "Lessons learned from the chameleon testbed," in *2020 USENIX Annual Technical Conference, USENIX ATC 2020, July 15-17, 2020*, pp. 219–233, USENIX Association, 2020.

[33] CWL: Common Workflow Language [online] https://www.commonwl.org/v1.2/, [Last accessed 26 August 2022].

[34] CWL Tools [online] https://github.com/common-workflow-language/cwltool

[35] CWL Airflow [online] https://github.com/Barski-lab/cwl-airflow

[36] Streamflow [online] https://streamflow.di.unito.it/

[37] Galaxy Project and portal [online] https://galaxyproject.org/

[38] S. Gallenmüller, D. Scholz, H. Stubbe, and G. Carle, "The pos framework: a methodology and toolchain for reproducible network experiments," in *CoNEXT '21: The 17th Int Conf on emerging Networking EXperiments and Technologies, Munich, Germany, December 7 - 10, 2021* (G. Carle and J. Ott, eds.), pp. 259–266, ACM, 2021.

[39] S. Gallenmüller, A pos Experiment in a Jupyter-Notebook https://github.com/gallenmu/pos-jupyter/blob/main/pos-experiment.ipynb

[40] JSON for Linking Data (JSON-LD) [online]

[41] Apache Atlas https://github.com/apache/atlas

[42] Amundsen https://github.com/amundsen-io/amundsen

[43] OpenMetadata https://open-metadata.org/

[44] EOSC Metadata Schema and Crosswalk Registry (MSCR) [online] https://faircore4eosc.eu/eosc-core-components/metadata-schema-and-crosswalk-registry-mscr

[45] Metadata extraction from code. Software Heritage [online] https://docs.softwareheritage.org/devel/architecture/metadata.html

[46] Stephan Druskat, et al, Software publications with rich metadata: state of the art, automated workflows and HERMES concept, Arxiv, 22 jan 2022 [online] Schlauch https://arxiv.org/abs/2201.09015