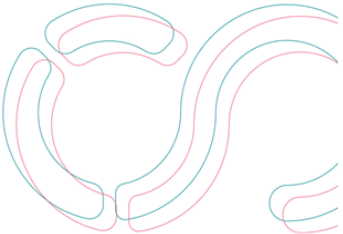
Infrastructures for Quality
Research Software Task Force

SIRS Gap Analysis Report

By the subgroup on the Information Science Perspective of the EOSC Task Force “Infrastructure for quality research software”:
Maxence Azzouz-Thuderoz, Laura Del Cano, Leyla Jael Castro, Łukasz Dumiszewski, Daniel Garijo, Jose Benito Gonzalez Lopez, Morane Gruenpeter, Moritz Schubotz, Marcin Wolski

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931



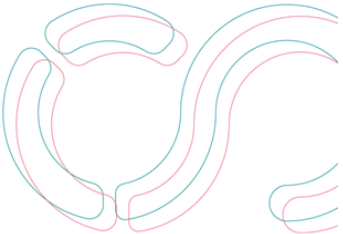
Executive Summary

The European Open Science Cloud (EOSC) Scholarly Infrastructures of Research Software (SIRS) report states that software is a necessary component of modern scholarly research and that EOSC has a key role to play, ensuring the overall architecture will be built in a way to cater the needs of the research community best. The SIRS report summarised key recommendations and technical requirements for the scholarly ecosystem to provide better functionalities and solutions for researchers. The SIRS recommendations stand on four pillars: Archive, Reference, Describe, and Cite research software. The SIRS report includes recommendations on archiving software artifacts, providing or exposing extrinsic and intrinsic identifiers, describing software with metadata, producing reliable citations, and improving the software records' curation quality across the European Open Science community.

This report reviews existing scholarly infrastructures that cater to research software and identifies the gaps between the EOSC Scholarly Infrastructures of Research Software (SIRS) desiderata and the status quo based on select examples.
In short, while there is a significant uptake of the SIRS recommendations, the development of a viable research software ecosystem in Europe is not currently foreseeable. Unlike the research data landscape, research software infrastructures are lagging behind. While there exists a highly reliable solution for archiving research software source code, not all source code is being adequately archived.

To improve this, policymakers should require proper archiving of software developed in publicly funded projects. Moreover, aggregators and publishers might be incentivized by establishing a certificate that ensures that software deposits are properly archived. Solutions to Reference, Describe, and Cite research software are less mature. Here the difference between research data and research software becomes more apparent. While many communities and institutions have reached a consensus on standards for managing research data and have established clear roles and responsibilities to ensure proper storage, description, referencing, and citation of such data, there is currently a lack of corresponding efforts for research software. Although most analysed infrastructures possess the technical capabilities to fulfil their roles as outlined in the SIRS reports, only a few are adequately prepared for the curation process in the event that a significant number of researchers deposit software or utilise references to research software.

Our analysis identifies concrete actions that the EOSC Infrastructures for Quality Research Software Task Force should support in order for scholarly infrastructures to close these gaps and comply with the SIRS recommendations. For example, we identified the gap that there exists a lack of common identifiers to reference software, and we suggest building an ecosystem around SWHIDs and establish them as a standard reference.



Tables of content

EOSC Association AISBL
Rue du Luxembourg 3, BE-1000 Brussels, Belgium
+32 2 537 73 18 | info@eosc.eu | www.eosc.eu
Reg. number: 0755 723 931 | VAT number: BE0755 723 931

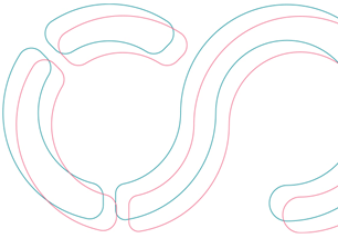
1. Introduction

In 2015, the European Commission launched the European Open Science Cloud[1], a major program[2] to unify, coordinate and monitor Open Science policies, actors, projects, and platforms in the European Union. To fulfil its promise to lead Europe towards Open Science, the EOSC Architecture Working Group (WG) launched the Task Force Scholarly Infrastructure for Research Software in 2021 with the mandate to offer recommendations on this topic. The output of this group was the Scholarly Infrastructures for Research Software report[3], which "suggests best practices, identifies open problems and describes use cases" to address the necessary further developments needed to build resilient infrastructures and services for the management of research software.

This document summarises the collaborative analysis carried out by the Working group on the Information Science Perspective (sub-group 2) of the EOSC Task Force Infrastructure for Quality Research Software. The analysis focused on evaluating the level of conformity of selected major resources used for archiving, describing, or storing research software to the recommendations outlined in the SIRS report. Additionally, the document outlines the actionable steps necessary to enhance their conformance to the report's recommendations.

The analysis focused on the gap between the recommendations from the SIRS report and what is currently provided by the available scholarly infrastructures for research software. The first part of this document highlights the gaps between the SIRS report requirements for *archival* of software and what is currently set among scholarly repositories. Then we come to the part dedicated to the *reference* guidelines, where we point out gaps in the available metadata schemata formats and how it will matter to have properly declared intrinsic and extrinsic identifiers. After that, we identify what still needs to be done to provide relevant *descriptions* for software and standards to be adopted for metadata schemata. In the last part, we report on the need for more adoption of tools and best practices to *credit and cite* research software authors.

[1] European Commission, "European Open Science Cloud (EOSC)."
[2] Commission, "European Cloud Initiative-Building a Competitive Data and Knowledge Economy in Europe."
[3] European Commission. Directorate General for Research and Innovation., *Scholarly Infrastructures for Research Software*.

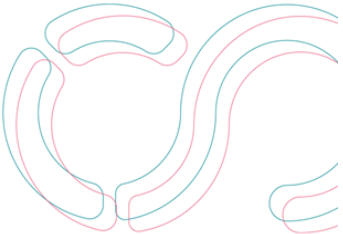


2. Methodology

The gap analysis reported in this document was carried out by a subgroup of the EOSC Task Force on Infrastructures for Quality Research Software. The Working Group included both researchers with expertise in research software and representatives of several infrastructures (scholarly repositories, tool registries, publishing platforms, etc.) analysed as part of the report.
The Working Group activity was organised in monthly meetings, led by two co-chairs, who organised the discussions and tracked the progress of the analysis report. The report was structured by filling a table with the main requirements for Research Software infrastructure addressed in the SIRS report for each resource included in the gap analysis. Each requirement corresponds to a row in the table, while each resource corresponds to a column. We define as resources the set of European infrastructures and metadata schemas having been analysed. The group decided to use six different labels to indicate the level of compliance of the resource to a given requirement:

+	“complies”: The target resource satisfies the requirement
*	“complies only partially”: The target resource partially satisfies the requirement
WIP	“work in progress”: The target resource is implementing mechanisms to satisfy this requirement
M	“missing”: The target resource currently does not satisfy this requirement
N/V	“needs to be verified”: The annotator needs to double-check or verify the compliance of this requirement
N/A	“not applicable”: The requirement does not apply to the analysed resource

At least one working group participant was assigned to each resource based on their familiarity with them (e.g., the participant was representing that resource within the Working Group) or their expertise (e.g., the participant had used the resource to deposit, publish or describe scholarly outputs).
Once a table was fully completed, the Working Group assigned participants to summarise each required category for all resources. It iterated in meetings until converging on a set of conclusions and future steps. The final result can be seen at the end of the document.



3. Analysed infrastructures

To assess the current level of compliance with the requirements outlined in the SIRS report, the Working Group identified and analysed a set of resources from the following categories outlined below. Most of the included resources have a general domain purpose, i.e., they are not explicitly targeted toward research software:

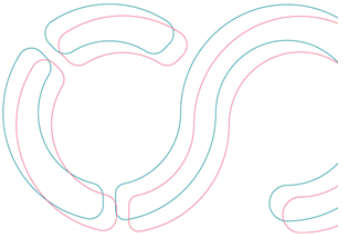
- **Scholarly repositories**: infrastructures where researchers and students can deposit their research outputs and all the research objects used to obtain such results;
- **Repository technologies**: software tools that can be instantiated and deployed to create scholarly repositories;
- **Publishing platforms**: infrastructures associated with journals or publishing companies that are used to deposit scholarly publications;
- **Aggregators**: infrastructures that register scholarly output metadata, and not necessarily the scholarly output themselves. The aggregation may be manual or automatic;
- **Aggregator technologies**: software tools that can be instantiated and deployed to create scholarly resource catalogues;
- **Metadata schemas**: frameworks defining how metadata (information providing details about an entity) should be organised and represented in a system thus to favour their exchange; ;

At least one resource/infrastructure from each category is included in the gap analysis. As for the inclusion criteria, resources were chosen for the following three main reasons:

1. The resource addresses the needs of a target community (e.g., mathematics);
2. A representative of the target resource was a member of the Working Group and volunteered to assess its level of compliance with the SIRS requirements; and
3. The resource or infrastructure is popular within a scientific community.

The resulting selection, which has no pretension of completeness, provides an illustrative sample of the resources and infrastructures in the European scene. The selection was based on the knowledge of the participants in the Task Force.

Appendix A.1 shows the list of resources included in our gap analysis and their description and category.



4. Gap Analysis

In this section, we go through the requirements and analyse the above-mentioned resources to determine their current level of conformance to the SIRS report recommendations and what would be needed to help them reach full conformance.

We split this section according to the general requirements for Archive, Reference, Describe, Cite/Credit, Easing Adoption, and Exemplarity Criteria. Each section will contain three sub-sections: a summary of the expectations from the SIRS report, the findings from the gap analysis, and the identification of actions needed to conform fully.

4.1. Archive

SIRS Report Summary

To support the "reproducibility, verifiability, and reusability of research results," software artifacts and associated extrinsic metadata should be preserved in the long run. This includes the software source code as well as associated tools and libraries. While the software source code, including self-describing metadata, is typically versioned, the individual versions are immutable. However, their associated extrinsic metadata are subject to moderation and editing.
To build and maintain the history of research software, the universal software archive and scholarly repositories need to interact to preserve the software in the context it was used originally. In particular, the SIRS report lists the following requirements/recommendations:

A1	Explicit deposit by identified individuals of one or more of the following: - software bundles with associated extrinsic metadata - extrinsic metadata associated with an artefact already existing in the universal archive
A2	Support non-public deposits and/or embargo periods
A3	Editing of extrinsic metadata
A4	(optional) Moderation of extrinsic metadata
A5	Download of the deposited bundle (as-is) and the associated metadata
A6	Repositories MUST feed the universal archive
A7	Repositories SHOULD keep a local copy
A8	Universal archive MUST keep track of the origin of the deposit
A9	Universal archive MUST provide provenance information to the repository

Gap analysis

Our gap analysis focuses on the capabilities of repositories rather than the capabilities and long-term sustainability of the universal archive.

Software Heritage, the universal source code archive, offers the possibility to infrastructures to deposit source code and metadata, which includes the name of the repository responsible for the submission, the time of the deposit, and extrinsic metadata[4][5]. An inherent Software Heritage identifier (SWHID) of the content will be generated using a cryptographic hash function, which prevents the content from being modified after the identifier was created. Moreover, it ensures that if the same code is deposited twice the same SWHID is used.

To the best of our knowledge, all the scholarly repository platforms allow the deposit of software bundles and their extrinsic metadata. Publishing platforms and aggregators also fully support extrinsic metadata. Non-public deposits and embargo periods are also supported. Our infrastructure set is compliant with SIRS recommendations about deposit functionalities.
Additionally, a substantial part of the analysed resources allows editing and downloading of software bundles and associated metadata, with few exceptions concerning downloading functionalities for Digital CSIC. We are pleased to note that most of the publishing platforms (e.g. Episciences, IPOL) have successfully incorporated the inclusion of a moderator role. This role serves to ensure the validity and accuracy of the metadata associated with the software. This integration of a moderator role enhances the quality and reliability of the content associated with the research software across these infrastructures.

All repositories maintain local copies of the source code; however, they may have varying strategies on how the Universal Software Archive is populated or fed with content. While RepOD acknowledges the importance of ensuring the archiving of source code in Software Heritage, KU Leuven RDR holds a different perspective and deems it unnecessary. This is because KU Leuven has established its own infrastructure to ensure the long-term archiving of software, but on the other hand, that software is not always findable and openly accessible. Zenodo, Dagstuhl, and swMATH are actively involved in the implementation of processes to feed content into the universal archive, primarily in the context of the FAIRCORE4EOSC[6] project. Furthermore, Digital CSIC has not yet started to work on feeding the universal archive.

In conclusion, the vast majority of repositories do not feed the universal archive yet. Currently, a limited number of initiatives are in the process of implementing the feeding mechanism for the universal archive. However, we believe that a broader audience can be persuaded to join these early adopters and actively participate in advancing this path. Technically, the development to feed the universal archive is doable since Software Heritage has developed APIs and endpoints to ease

[4] Di Cosmo, Gruenpeter, and Zacchiroli, “Referencing Source Code Artifacts.”
[5] Cosmo, Gruenpeter, and Zacchiroli, “204.4 Identifiers for Digital Objects.”
[6] European Union, “Core Components Supporting a FAIR EOSC.”

software archiving. In particular, the extent to which these platforms and infrastructures recognise the value of open sourcing their software to enhance visibility, as well as the importance of storing them in a universal archive accessible to all, holds paramount significance in driving the decision to embrace the recommendations outlined in the SIRS report.

4.2. Reference

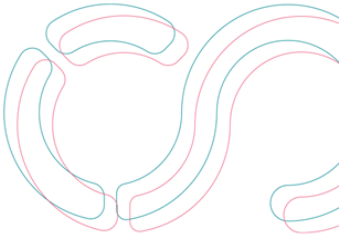
SIRS Report Summary

Support reproducibility and verifiability of research results ensuring unambiguous identification of a software artefact and/or the associated metadata. Concretely, supporting *Intrinsic Identifiers* (decentralised and specifically designed for software source code) and *Extrinsic Identifiers* (register-based, editing of the metadata associated with a deposit without changing the identifier compatibility with the traditional workflow in the scholarly ecosystem).
More concretely, the SIRS report lists the following requirements/recommendations:

R1	Intrinsic identifiers
R2	Extrinsic identifiers
R3	All references to a publicly available software artefact MUST include a qualified intrinsic identifier; references to a non-publicly available software artefact SHOULD include an intrinsic identifier.
R4	References to research software artefacts that are explicitly deposited in a scholarly repository MUST include the corresponding extrinsic identifier.
R5	References to software projects that are not software artefacts MUST include a qualified extrinsic identifier.

Gap analysis

Only a few services, e.g. HAL, comply with the SIRS requirements regarding the Referencing software. Most infrastructures in this study do not support intrinsic identifiers at all. A few others support adding intrinsic identifiers by reference (i.e. extra metadata with alternative identifiers) and not tightly integrated with the Universal Archive (Software Heritage) to use SWHIDs. Further work is needed in enabling SWHIDs in the current Scholarly Infrastructure, and interconnecting all components of the infrastructure with the Universal archive, as shown in the figure below. Such interconnection would facilitate the creation and exchange of identifiers (both intrinsic and extrinsic) in a seamless and transparent way for the researchers.



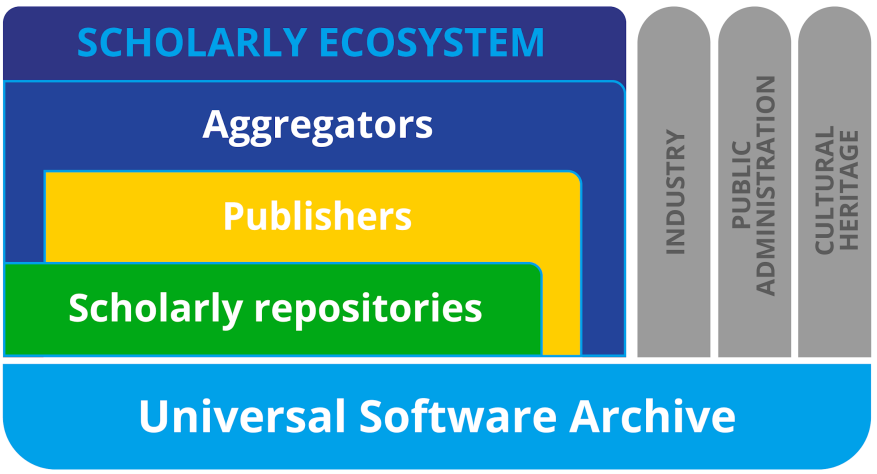


Figure 1: Recommended architecture by the SIRS report

E*xtrinsic identifiers* (e.g. DOI, Handle, etc) seem to be widely supported across all types of infrastructures (repositories, aggregators, publishers) and they are well integrated into the whole publication workflow.

Based on the results from the gap analysis, *references to public and non-public available software artefacts with an associated intrinsic and extrinsic identifier* seems to be not well understood or lacking. This requirement is targeted at publishing platforms and authors as well as scholarly repositories. This means scholarly repositories which already provide extrinsic identifiers should support software metadata containing intrinsic identifiers. In this way, whenever an author refers to software, an intrinsic and/or an extrinsic identifier must be used to reference it. Only a few publishers have already explicitly integrated this recommendation, like IPOL, eLife, JTCAM, and the journals hosted on the Episcience platform. The publishing platform Dagsthul is working to support intrinsic identifiers like the SWHID as well as the aggregator swMATH within the FAIRCORE4EOSC project. The scholarly repository HAL and the publishing platform IPOL already support this intrinsic identifier.

Measures should be taken to increase the uptake of software identifiers used in publications. While there are clear guidelines on how software should be cited[7], compliance is not yet enforced. Incentives for publishers and aggregators are needed to facilitate the adoption of the new referencing culture for software. The challenge is both technical[8] and cultural. Technical, as making software metadata easily findable and citation formats downloadables can help to have scholars properly citing software but also cultural, as still a lot of researchers continue to cite papers presenting a software and not the software itself.

[7] Smith et al., “Software Citation Principles.”
[8] Niemeyer, Smith, and Katz, “The Challenge and Promise of Software Citation for Credit, Identification, Discovery, and Reuse.”

4.3. Describe

SIRS Report Summary

Metadata plays a significant role in the proper discoverability of research software artefacts. The intrinsic metadata (i.e. metadata found in the source code) is foreseen as a primary means for managing the software-specific information.
To conveniently describe software content three main elements have to be properly set up: 1) vocabularies and ontologies (such as the extension of schema.org), 2) tools to create, edit, validate, and convert the metadata, and 3) registries to store the metadata. Furthermore, publishers and scholarly repositories should play a prominent role in the diffusion of software description information. Publishers should be obliged to equip the publications with proper metadata about software, while the repositories provide the necessary means for metadata curation.
Another crucial aspect is metadata interoperability i.e. the ability to exchange software metadata among federated infrastructures. To achieve the relevant Interoperability level several requirements must be met. The metadata must be available in a machine-readable format and must be compliant with standard vocabularies to ensure its effective sharing. Metadata should support versioning, relations (such as relations with external identifiers like DOI), and proper binding with other types of scientific resources (such as publications). The intrinsic metadata should be stored in compliance with software best practices. Here are the SIRS report recommendations regarding software metadata description (See Appendix A.1 - Table 2):

D1	Intrinsic metadata: found in the source code itself
D2	Extrinsic metadata: created via a deposit, publication, or aggregation process
D3	Vocabularies and ontologies
D4	Tools to create, edit, validate, and convert metadata
D5	Registries to store metadata
D6	Metadata MUST be made available in a machine-readable form using a standard vocabulary adapted for software. CodeMeta (Jones et al., 2016) is a good candidate
D7	Intrinsic metadata MUST be created and stored according to recognised best practices in software development
D8	Metadata SHOULD support relations: versioning (part of same software, new version, etc)
D9	Metadata SHOULD support relations: relations with other research objects (papers, etc)
D10	Metadata SHOULD support relations: relations with other identifiers (DOI vs SWHID)
D11	Information specific to a software artifact SHOULD be in the intrinsic metadata

D12	Publishers MUST ensure that software associated with the publication is equipped with proper metadata
D13	Scholarly repositories SHOULD provide the necessary means to support metadata curation

Gap analysis

Following the SIRS report, we have assessed the extent to which each analysed resource deals with research software metadata. The lack of means or efforts to curate intrinsic metadata in software artefacts is salient. Only two resources (HAL and Digital-CSIC) support the extraction of intrinsic metadata, found in the source code itself. Inversely, noticeable efforts can be highlighted to properly curate extrinsic metadata. These unbalanced efforts can lead in the long run to a serious deterioration in the global information quality of the software. Efforts to properly store these intrinsic metadata are also not often done, as software development good practices are not generalised yet in the scholarly world. Digital CSIC is certainly the infrastructure caring the most about the quality of the information describing software artefacts, as they strongly support good practices by providing scholars with README templates to ensure edited intrinsic metadata are reliable.

Most of the resources include tools for editing, creating, or validating extrinsic metadata. Such tools are often supplied along the depositing, publication, or aggregation process.

Most repositories, publishing platforms as well as aggregators support a bunch of usual metadata schemas, but with a few designed for software. Even if some platform has already adopted software metadata vocabularies or plans to adopt it, many of the scholarly repositories have an emphasis on data and not on software, explaining why so much does not have planned yet to support it. This lack of software concern is also a reason why so few support ontologies describing software.

Thus, we believe software metadata vocabularies adoption to be a point of improvement for the resources in our analysis, to follow best practices proposed by the community (Garijo et al., 2022) and provide interoperability by adopting (or extending) standard vocabularies like Codemeta, as proposed in the SIRS report. Currently, only a few infrastructures support CodeMeta. Scholarly Repositories such as eCienciaDatos, KULeuven, RepOD, and Digital CSIC have not yet followed the SIRS report recommendations as well as the publishing platform EMS Press and the aggregator OpenAIRE.
Nevertheless, a lot of them already support Schema.org vocabulary which is the parent vocabulary of Codemeta. Such an adoption could be certainly achieved with few efforts.
However, the aggregator bio.tools already supports the Bioschemas computational tool format, with a vocabulary derived from Schema.org, and is so much compatible with CodeMeta[9]. Thus, we do not consider it necessary for this platform to support CodeMeta.

[9] Jones et al., “CodeMeta.”

Also, our analysis indicates almost all infrastructures support metadata versioning, relations with other research objects, or other identifiers. For each new version of the software, users can access the associated metadata of the specified version. Some work is still needed for the repository eCienciaDatos and publishers EMS Press and Dagshtul to comply with these recommendations.

Publishers must ensure that software associated with the publication is equipped with proper metadata, but as far as we know, Dagsthul is the only publisher already imposing it on users, while EMS Press, IPOL, and Episciences are currently working to implement it.

Almost all scholarly repositories provide the necessary means to support assisted metadata, only Zenodo is still working to implement it.

4.4. Cite/Credit

SIRS Report Summary

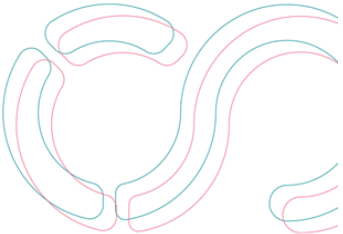
The SIRS report describes the need for an initial classification of contributor roles for research software based on a decade-long experience at INRIA and CNRS. These roles are architecture, coding, debugging, design, documentation, maintenance, management, support, and testing. Other requirements outlined by the SIRS report include the following needs (See Appendix A.1 - Table 2):

C1	Classification of contributor roles for research software
C2	Bibliographic citation data model adapted for software.This is the subset of software metadata needed for producing a citation in all contexts of interest.
C3	Machine readable representation of the data model (BibTex, etc)
C4	Citation styles for typesetting the citation data
C5	Plagiarism detection mechanisms
C6	(optional) Expert peer evaluation

Gap analysis

Credit in Research Software

The adoption of contribution roles is not yet a common research practice - only covered partially by 5 resources (HAL, KU Leuven RDR, Digital CSIC, InvenioRDM) out of 18. This is true not only for



software but also for data and, to a lesser extent, scholarly articles. In the case of scholarly articles, it is a common practice across some publishers to specify the sort of contributions made by each author; however, this information is commonly text-based and not part of the metadata shared by publishers and aggregators, i.e., scholarly article repositories, either on their websites or via bulk downloads. Controlled vocabularies such as the Contributors Roles Taxonomy (CRediT, https://credit.niso.org/) and the Contributor Role Ontology (CRO, https://obofoundry.org/ontology/cro.html) are two promising projects for the recognition of different roles that is improving and becoming a more common practice. Still, more work is needed to better credit work related to software. CRediT, which is becoming a popular choice for scholarly articles, includes a generic role as "software" while CRO offers is better suited to software as it recognizes contributions to architecture, design, engineering, and testing. Even metadata schemas such as CodeMeta and Bioschemas do not provide clear support to contribution roles, although they could be easily included (e.g., using the role approach defined in schema.org and combining it with a contribution role vocabulary). Schemas mostly distinguish authors from contributors; however, the coverage of these two categories is not clear.

Citation in Research Software

Bibliographic citation models, both for data and software, follow the same approach used for scholarly articles. Two of the analysed repositories (HAL and Zenodo) propose different models for software citations, like the one of BibTex[10] or DataCite[11]. However, the adoption of community citation models such as the Citation File Format[12] (CFF) is not yet commonplace among the analysed infrastructures. CFF is a model commonly adopted among research software engineers and researchers producing software and designed to be integrated into GitHub and GitLab repositories. GitHub, for instance, supports rendering CFF files on code repositories' landing pages as a BibTex snippet (since the adoption of CFF, thousands of repositories have started to use them on the platform). Several of the analysed repositories support citation styles for typesetting the citation data, to ease usability by users. But the adoption of specific software entries as the one proposed in the SIRS report is largely not yet achieved. It is also worth noting the biblatex-software package is yet broadly adopted for software citations among researchers but is only already available on HAL.

Adoption of Metadata Vocabularies

With regards to presenting metadata in a machine-readable way, the inclusion of structured markup on websites is becoming more and more popular in research thanks to efforts such as schema.org, Bioschemas, and CodeMeta. Despite the availability of supporting schemas (e.g., Codemeta at https://codemeta.github.io/ and Bioschemas Computational Tool at https://bioschemas.org/profiles/ComputationalTool/), providing machine-readable representation of the software citation model (even if not tailored to software) is not yet a common practice. However, there seems to be more progress in meeting this requirement than in meeting all the other credit/citation requirements, as about 50% of the analysed repositories do support machine-readability for the

[10] Patashnik, "BibTEXing."
[11] DataCite Metadata Working Group, "DataCite Metadata Schema Documentation for the Publication and Citation of Research Data and Other Research Outputs v4.4."
[12] Druskat, Stephan et al., "Citation File Format."

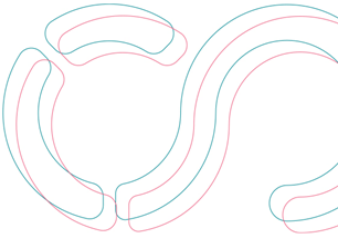
citation model. Initiatives such as the SciCodes consortium (https://scicodes.net/), the Research Software Directory (https://www.esciencecenter.nl/research-software-directory/), and bio.tools (https://bio.tools/) are slowly widening the adoption of best practices in registries. Most of the resources described in this report support functionalities to edit software metadata to leave the possibility to users to download them in different machine-readable formats (e.g., BibTex), with noticeable exceptions for swMATH and bio.tools. Furthermore, they also include the possibility for scholars to ingest metadata in a given format to convert them into other vocabulary styles.

Plagiarism detection

Plagiarism detection mechanisms are not supported by any repository or aggregator included in our gap analysis. One of the challenges here is the evaluation of the number of pieces of software shared in, for example, tutorials or software question/answer websites with a data licence or no licence at all. Some RSEs and researchers opt for copying the code and adding a note in the form of a comment to somehow credit the original authors or post. There is a need to introduce a clear and appropriate definition of plagiarism.

Peer review support by scholarly repositories

One publishing platform (IPOL) fully supports peer evaluation while one repository (Digital CSIC) and one aggregator offer partial support (Géant Software Catalogue) for it. Also, we wish to distinguish the research software peer evaluation process, which consists to validate that the software content corresponds well to what is presented in the research article, from the traditional programming peer review that we usually find in the industry which focuses on the code quality. Nonetheless, expert peer validation remains costly, and hence not many institutions can afford it.



5. Exemplarity Criteria for Participating Infrastructures

While the four previous sections, Archive, Reference, Describe, and Credit, refer to those scholarly activities that should be supported by scholarly infrastructures for research software, in this section we focus on exemplarity criteria for funding open scholarly infrastructures, mostly concerning openness and availability of the metadata associated with the research software archived by participating infrastructures.

5.1. Openness

SIRS Report Summary

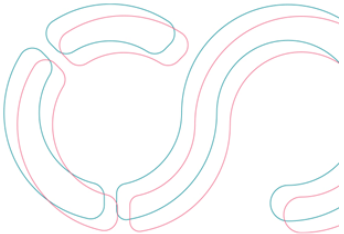
The SIRS report highlights the importance of openness for software metadata with the emphasis on infrastructures and means to access them:

O1	Metadata should be accessible in a standard format and under a CC0 licence
O2	Access to the metadata and the data should be possible through an open API using standard protocols and without identification
O3	Aggregated metadata should be available “as open as possible as closed as necessary”
O4	The infrastructures should be built from stable existing open source software building blocks, and all the software of the infrastructure should be available under an open source licence
O5	Communications and data exchange use open standards for data formats and protocols
O6	The infrastructure should be hosted and run by a non-profit organisation to avoid risk of proprietarisation

Gap analysis

The analysed repositories and repository technologies meet all the criteria corresponding to Openness, with two exceptions: eCienciaDatos misses the documentation of its standard format and open API, while FigShare is neither hosted nor run by a non-profit organization. Similarly, the publishing platforms meet all the Openness requirements (Episciences) or are currently adapting to them (EMS Press).

There are different levels of meeting the Openness requirements by the aggregators included in our analysis. HAL, RepOD, KU Leuven RDR, EpiSciences, IPOL, and OpenAIRE fully meet all the criteria. swMATH does not use an open standard format nor an open API for accessing metadata, and it only partially meets 1) the criteria of open source software used to build infrastructure and 2) the



open standard of data exchange. swMATH fully meets all the other requirements regarding Openness. Géant Software Catalogue does not use standard formats for its metadata and data exchange; however, there is ongoing work to support public projects and provide an open API that could ease access to the associated metadata. Dagsthul restricts its API to its users and does not support the licence CC0 for its metadata as well as Digital CSIC. The adoption of the licence CC0 remains a challenge beyond these two infrastructures, even if adopted licences by other infrastructures are not much restrictive.

5.2. Governance

SIRS Report Summary

In the report, authors come over with what could be seen as a guideline of management practice of people to establish clear open science policies for institutions involved in this process:

G1	Clear definition of governance bodies
G2	Procedures for the selection of governance bodies' members are clearly and publicly stated
G3	Procedures for participation are clearly and publicly stated

Gap analysis

It seems that a clear description of governance bodies is still missing for various infrastructures, especially, or difficult to find. Procedures of participation and selection are even more rare. Infrastructures should put an effort on this open governance practices.

5.3. Sustainability

SIRS Report Summar/y

Financial and technical considerations in the long run to ensure software preservation are also present in the report:

S1	The general operation of the infrastructure or platform is not based on the financing of one-off projects
S2	A plan for long term availability of the service exists and is made public
S3	An exit strategy that could give continuity to the data and metadata beyond the life of the service

Gap analysis

In general, it seems that sustainability of services is beyond one-off projects, but there are some important repositories and standards in which this is not the case (Bioschemas, Zenodo, IPOL, EpiScience, swMATH) and depend on such funding. In some cases, the long-term commitment is limited to running the technical infrastructure but missing further development of the service. Besides, it is important to highlight that it is not easy to find sustainability information online for most resources (Digital CSIC for instance).
On the technologies (e.g. InvenioRDM) and standards (e.g. CodeMeta) side, many of them are community-driven or FOSS initiatives that do not have long-term sustainability funding, although they might have a medium-term public plan.
eCienciaDatos, Digital CSIC, HAL and Episcience have set a strategy in case of service decommissioning, which represents a risk for long-term sustainability. However, only eCienciaDatos made it publicly available.

5.4. Transparency

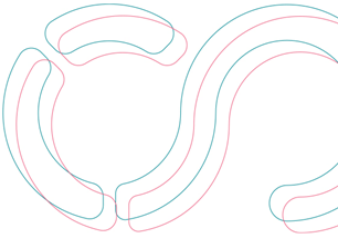
SIRS Report Summary

The last criteria presented by authors of the SIRS report are transparency both for the users and the financial sources:

T1	Terms of use are clearly and publicly stated
T2	Sources of funding are clearly and publicly stated

Our gap analysis

Almost all the analysed resources state on their websites the terms of use and sources of funding, except for Episciences and IPOL.



6. Summary and Recommendations

In this report, we have conducted a gap analysis of the recommendations of the SIRS report over 18 resources who support functionalities for the management of research software. On the one hand, we found how some infrastructures have already implemented good open science practices, as reported in the ARDC principles. In particular, the **archiving**, **transparency**, and **openness** recommendations **have been adopted** by most of the infrastructures in our study. On the other hand, recommendations regarding governance, credit, sustainability, and metadata still require significant work for adoption. We summarise the main findings of our study below, along with recommended action points to address them:

6.1. Archive

A few scholarly infrastructures are involved in feeding the universal archive. Metadata should be editable and deposited by platforms in the universal archive to find source code and metadata in the same place. However, this is not possible yet.
It is worth noticing that some scholarly repositories like Zenodo or HAL are often archiving software from general software repositories like GitHub or GitLab. For these last ones, the Software Heritage webhook can also be easily used to trigger source code archiving. Beyond the efforts of scholarly repositories, publishers, or aggregators, the good practices among scholars regarding software archiving are also crucial.
To guarantee that the archiving process preserves the integrity of software bundles and their associated metadata, scholarly repositories must align with the universal archive's technical specifications.

6.2. Reference

Referencing software remains a challenge; there has been an uptake for extrinsic identifiers. However, very few platforms use compatible intrinsic identifiers for their software entries. The usage of SWHIDs and the interconnection between scholarly repositories and the universal archive is rising slowly. Further work is needed to **enable Software Heritage IDs (SWHIDs) in the current Scholarly Infrastructure** and interconnect all infrastructure components with the Universal archive. Requiring using SWHIDs in software sustainability plans for project proposals may help improve the adoption of the SIRS recommendations.

6.3. Describe

Recommendations for describing research software show a mixed level of adoption. While most platforms propose tools to edit the software metadata of their entries, only some extract intrinsic values from documentation and code, hence pushing the metadata curation problem toward software authors. Support for intrinsic metadata is currently implemented by a minority of existing platforms and only in a limited scope. The adoption of guidelines and common **software quality indicators is**

needed to assist researchers in providing the key metadata for others to use their software components.
In addition, current infrastructures still **need to adopt a *standard software metadata schema***, such as CodeMeta, for representing and interchanging software entries. Aggregators **rarely provide access to the metadata and data through an open API** using standard formats and protocols and under a CC0 license. We believe addressing these points is crucial for the adoption of the SIRS recommendations and compliance with the FAIR principles, e.g., by funding efforts to ensure community governance in CodeMeta and having incentives for interchanging software metadata.

6.4. Cite/Credit

The roles of developers in research software are poorly represented within current infrastructures. CRediT[13], a common taxonomy used to represent roles in research, has a single category for software, making it difficult to distinguish who contributed in the different stages of the software development process. Current metadata schemas mostly distinguish authors from contributors, without making a clear distinction of roles. A **research software contributor roles taxonomy is needed** to provide proper credit to software developers. Contribution Role Ontology (CRO) is an ongoing work in CodeMeta, and HAL has integrated the 9 roles for software developers identified in the CiSE 2020 survey[14].

As for software citation, while the scientific community has produced good practices and standards these are yet **not widely adopted in the analysed infrastructures**. We believe this is a work in progress for some infrastructures, and we expect standards like Codemeta and CFF to be widely adopted in the future.

Other mechanisms related to credit for software projects, such as plagiarism detection, peer review, or usage analytics are not widely adopted so far. The **adoption of common software quality indicators** and **best practices** for authors may help partially automate aspects of peer review while providing insightful metrics to software authors and funders.

6.5. Governance and sustainability

It is worth noting sustainability seems not to be a real issue except for a few platforms, and that transparency of relevant information about it could be certainly improved. However, governance rules suffer from a lack of clear definitions, and improvement appears like a necessity to ensure their long-term viability. We believe that these aspects should be included in the infrastructure's **software management plan** when applying for funding, and clearly stated on the infrastructure's website following community best practices[15].

[13] Allen, O'Connell, and Kiermer, "How Can We Ensure Visibility and Diversity in Research Contributions?"
[14] Alliez et al., "Attributing and Referencing (Research) Software."
[15] Garijo et al., "Nine Best Practices for Research Software Registries and Repositories."

In addition, we believe that **supporting an EOSC Working Group to engage with the main publishers** is required to make the SIRS recommendations a reality. Engagement should happen at two different levels: 1) Technical, e.g., representing software within the Journal Article Tag Suite (JATS) format[16]; and 2) Social, by developing best practices for authors citing and using the software within their publications.

Table 1. The table below summarises our gap analysis and proposed action points:

Dimension	**Gap**	**Action point**
Archive	Software source codes not archived	Using the SWH API to ensure source code archiving
Archive	Metadata describing software not archived	Using the SWH API to ensure metadata archiving
Reference	Lack of common identifiers for software	Requiring using SWHIDs in software sustainability plans for project proposals
Describe	Intrinsic metadata support	Adoption of guidelines and common software metadata quality indicators is needed
Describe	Lack the adoption of a *common software metadata schema*	Encourage use of CodeMeta, funding for community and governance
Describe	Aggregators rarely provide access to the metadata and data through an open AP	
Credit	No research software contributor roles taxonomy	Generalisation of Contribution Role Ontology (CRO) and integration in Codemeta.
Credit	No common software citation practice adopted in infrastructures	Require adoption of standards like CFF.
Credit	No common software quality indicators and best practices for software metadata	Development of common software indicators, guides based on Codemeta
Governance and sustainability	No adoption of SIRS recommendations from infrastructures	Require governance and sustainability to be part of the software management plan. Compliance with best community practices
All	Adoption of SIRS recommendations by publishers	Working Group to synchronise and ease the adoption of community standards into JATS

[16] NISO JATS Standing Committee, “ANSI/NISO Z39.96-2021, JATS.”

7. Acknowledgment

The subgroup on the Information Science Perspective of the EOSC Task Force "Infrastructure for quality research software" wants to express its gratitude to Estelle Nivault, Yannick Barborini, Isabel Bernal Martinez, Mark Hahnel, Krzysztof Madry, Dieuwertje Bloemen, Ramy-Badr Ahmed, Michael Wagner, Maud Medves, Miguel Colom, Maciej Łabędzki, Maaike de Jong and Jason Maassen for their significant contribution to this report. They were concerned with being as authentic and accurate as possible to identify where their infrastructure does not comply with the SIRS recommendations and find the possible actions to trigger to overcome these gaps.

8. References

Allen, Liz, Alison O'Connell, and Veronique Kiermer. "How Can We Ensure Visibility and Diversity in Research Contributions? How the Contributor Role Taxonomy (CRediT) Is Helping the Shift from Authorship to Contributorship." *Learned Publishing* 32, no. 1 (January 2019): 71–74. https://doi.org/10.1002/leap.1210.

Alliez, Pierre, Roberto Di Cosmo, Benjamin Guedj, Alain Girault, Mohand-Said Hacid, Arnaud Legrand, and Nicolas P. Rougier. "Attributing and Referencing (Research) Software: Best Practices and Outlook from Inria." *Computing in Science & Engineering* 22, no. 1 (January 1, 2020): 39–52. https://doi.org/10.1109/MCSE.2019.2949413.

Commission, European. "European Cloud Initiative-Building a Competitive Data and Knowledge Economy in Europe." *COM 2016*, 2016.

Cosmo, Roberto Di, Morane Gruenpeter, and Stefano Zacchiroli. "204.4 Identifiers for Digital Objects: The Case of Software Source Code Preservation.," September 21, 2018. https://doi.org/10.17605/OSF.IO/KDE56.

DataCite Metadata Working Group. "DataCite Metadata Schema Documentation for the Publication and Citation of Research Data and Other Research Outputs v4.4." Application/pdf, 2021, 82 pages. https://doi.org/10.14454/3W3Z-SA82.

Di Cosmo, Roberto, Morane Gruenpeter, and Stefano Zacchiroli. "Referencing Source Code Artifacts: A Separate Concern in Software Citation." *Computing in Science & Engineering* 22, no. 2 (March 2020): 33–43. https://doi.org/10.1109/MCSE.2019.2963148.

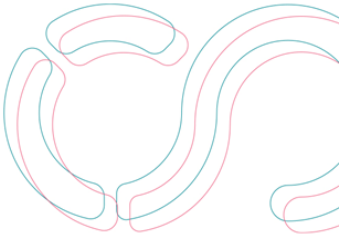
Druskat, Stephan, Spaaks, Jurriaan H., Chue Hong, Neil, Haines, Robert, Baker, James, Bliven, Spencer, Willighagen, Egon, Pérez-Suárez, David, and Konovalov, Alexander. "Citation File Format," August 9, 2021. https://doi.org/10.5281/ZENODO.1003149.

European Commission. Directorate General for Research and Innovation. *Scholarly Infrastructures for Research Software: Report from the EOSC Executive Board Working Group (WG) Architecture Task Force (TF) SIRS.* LU: Publications Office, 2020. https://data.europa.eu/doi/10.2777/28598.

European Union. "Core Components Supporting a FAIR EOSC," September 4, 2022. https://doi.org/10.3030/101057264.

Garijo, Daniel, Hervé Ménager, Lorraine Hwang, Ana Trisovic, Michael Hucka, Thomas Morrell, Alice Allen, Task Force on Best Practices for Software Registries, and SciCodes Consortium. "Nine Best Practices for Research Software Registries and Repositories." *PeerJ Computer Science* 8 (August 8, 2022): e1023. https://doi.org/10.7717/peerj-cs.1023.

Jones, Matthew B, Carl Boettiger, Abby Cabunoc Mayes, Arfon Smith, Peter Slaughter, Kyle Niemeyer, Yolanda Gil, et al. "CodeMeta: An Exchange Schema for Software Metadata." Application/ld+json. KNB Data Repository, 2017. https://doi.org/10.5063/SCHEMA/CODEMETA-2.0.



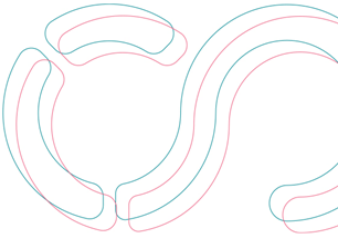
Lin, Dawei, Jonathan Crabtree, Ingrid Dillo, Robert R. Downs, Rorie Edmunds, David Giaretta, Marisa De Giusti, et al. “The TRUST Principles for Digital Repositories.” *Scientific Data* 7, no. 1 (May 14, 2020): 144. https://doi.org/10.1038/s41597-020-0486-7.

Niemeyer, Kyle E., Arfon M. Smith, and Daniel S. Katz. “The Challenge and Promise of Software Citation for Credit, Identification, Discovery, and Reuse.” *Journal of Data and Information Quality* 7, no. 4 (October 13, 2016): 1–5. https://doi.org/10.1145/2968452.

NISO JATS Standing Committee. “ANSI/NISO Z39.96-2021, JATS: Journal Article Tag Suite, Version 1.3.” NISO. Accessed March 16, 2023. https://doi.org/10.3789/ansi.niso.z39.96-2021.

Patashnik, Oren. “BibTEXing.” February, 1988.

Smith, Arfon M., Daniel S. Katz, Kyle E. Niemeyer, and FORCE11 Software Citation Working Group. “Software Citation Principles.” *PeerJ Computer Science* 2 (September 19, 2016): e86. https://doi.org/10.7717/peerj-cs.86.



9. Appendix A.1 Infrastructures

Table 1 provides short descriptions of the type and goals of all the resources analysed in the SIRS gap analysis report.

Table 1: List of all resources included the GAP analysis report

Resource	Short description	Type
Zenodo	Repository platform for archiving all types of scholarly resources	Scholarly Repository
HAL	Open archive where authors can deposit scholarly documents from all academic fields	Scholarly Repository
e-cienciaDatos	Multidisciplinary data repository that houses the scientific datasets of researchers from the public universities of the Community of Madrid (Spain), Universidad Nacional de Educación a Distancia (UNED), members of the Consorcio Madroño.	Scholarly Repository
ArXiv	Open-access repository of electronic preprints and postprints approved for posting after moderation, but not peer review.	Scholarly Repository
Figshare	Online open access repository for researchers to preserve and share their research outputs, including figures, datasets, images, and videos.	Scholarly Repository
RepOD	General-purpose repository for open research data in Poland. It is intended for scholarly data	Scholarly Repository

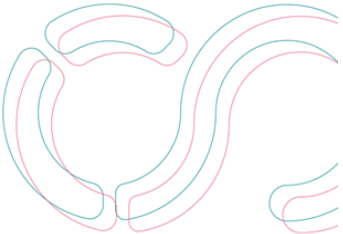
	from all disciplines of knowledge and in all formats.	
KU Leuven RDR	Institutional repository that helps KU Leuven researchers publish, share, cite, and preserve their research data in a findable, accessible, interoperable and reusable way.	Scholarly Repository
Digital CSIC	Online open access repository of research produced by the Spanish National Research Council	Scholarly Repository
Dataverse	Dataverse is an open source software platform for sharing, finding, citing, and preserving research data (developed by the Data Science and Products team at the Institute for Quantitative Social Science and the Dataverse community).	Repository technology
InvenioRDM	Repository application that anyone can use to run a service similar to e.g. Zenodo.	Repository technology
European Mathematical Society Press (EMS Press)	Publishing house of the European Mathematical Society, a not-for-profit organization dedicated to the promotion and development of mathematics in Europe.	Publishing platform
Dagstuhl Publishing	Dagstuhl's central task is to **enable and promote communication between researchers in computer science**.	Publishing platform

Episciences	Overlay journal platform, hosts and disseminates open access publications from all disciplines. Supported by the CCSD (The Center for Direct Scientific Communication) grouping the CNRS (The French National Centre for Scientific Research), the INRIA (National Institute for Research in Digital Science and Technology) and the INRAE (National Research Institute for Agriculture, Food and Environment).	Publishing platform
Image Processing On Line (IPOL)	Research journal of image processing and image analysis which emphasises the role of mathematics as a source for algorithm design and the reproducibility of the research.	Publishing platform
OpenAIRE	A network of Open Access repositories, archives and journals that support Open Access policies.	Aggregator
swMATH	swMATH is a freely accessible, innovative information service for mathematical software.	Aggregator
bio.tools	Registry storing essential scientific and technical information about software tools, databases and services for bioinformatics and the life sciences.	Aggregator
Research Software Directory	The Netherlands eScience Center uses the Research Software Directory to show our software packages and tools.	Aggregator

	The system is open source and is now available as a service.	
GÉANT Software Catalogue	Catalogue of software projects and teams, established in order to facilitate and support the GÉANT software engineering community and collect information about GÉANT software development efforts	Aggregator Technology
Codemeta	Vocabulary that can be used to standardise the exchange of software metadata across repositories and organisations.	Metadata schema
Bioschemas	Bioschemas aims to improve the Findability on the Web of life sciences resources such as datasets, software, and training materials.	Metadata schema

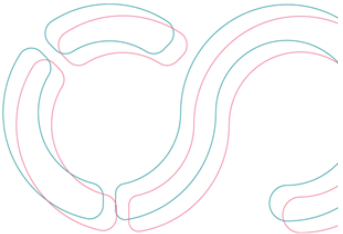
Table 2: The indexed SIRS recommendations used in the GAP analysis report

A1	Explicit deposit by identified individuals of one or more of the following: - software bundles with associated extrinsic metadata - extrinsic metadata associated with an artefact already existing in the universal archive
A2	Support non-public deposits and/or embargo periods
A3	Editing of extrinsic metadata
A4	(optional) Moderation of extrinsic metadata
A5	Download of the deposited bundle (as-is) and the associated metadata
A6	Repositories MUST feed the universal archive
A7	Repositories SHOULD keep a local copy
A8	Universal archive MUST keep track of the origin of the deposit
A9	Universal archive MUST provide provenance information to the repository
R1	Intrinsic identifiers
R2	Extrinsic identifiers
R3	All references to a publicly available software artifact MUST include a qualified intrinsic identifier; references to a non-publicly available software artifact SHOULD include an intrinsic identifier.
R4	References to research software artifacts that are explicitly deposited in a scholarly repository MUST include the corresponding extrinsic identifier.
R5	References to software projects that are not software artifacts MUST include a qualified extrinsic identifier.
D1	Intrinsic metadata: found in the source code itself
D2	Extrinsic metadata: created via a deposit, publication, or aggregation process
D3	Vocabularies and ontologies
D4	Tools to create, edit, validate, and convert metadata
D5	Registries to store metadata



D6	Metadata MUST be made available in a machine-readable form using a standard vocabulary adapted for software. CodeMeta (Jones et al., 2016) is a good candidate
D7	Intrinsic metadata MUST be created and stored according to recognised best practices in software development
D8	Metadata SHOULD support relations: versioning (part of same software, new version, etc)
D9	Metadata SHOULD support relations: relations with other research objects (papers, etc)
D10	Metadata SHOULD support relations: relations with other identifiers (DOI vs SWHID)
D11	Information specific to a software artifact SHOULD be in the intrinsic metadata
D12	Publishers MUST ensure that software associated with the publication is equipped with proper metadata
D13	Scholarly repositories SHOULD provide the necessary means to support metadata curation
C1	Classification of contributor roles for research software
C2	Bibliographic citation data model adapted for software.This is the subset of software metadata needed for producing a citation in all contexts of interest.
C3	Machine readable representation of the data model (BibTex, etc)
C4	Citation styles for typesetting the citation data
C5	Plagiarism detection mechanisms
C6	(optional) Expert peer evaluation
Exemplarity Criteria for Participating Infrastructures	
O1	Metadata should be accessible in a standard format and under a CC0 license
O2	Access to the metadata and the data should be possible through an open API using standard protocols and without identification

O3	Aggregated metadata should be available “as open as possible as closed as necessary”
O4	The infrastructures should be built from stable existing open source software building blocks, and all the software of the infrastructure should be available under an open source license
O5	Communications and data exchange use open standards for data formats and protocols
O6	The infrastructure should be hosted and run by a non-profit organisation to avoid risk of proprietarisation
G1	Clear definition of governance bodies
G2	Procedures for the selection of governance bodies’ members are clearly and publicly stated
G3	Procedures for participation are clearly and publicly stated
S1	The general operation of the infrastructure or platform is not based on the financing of one-off projects
S2	A plan for long term availability of the service exists and is made public
S3	An exit strategy that could give continuity to the data and metadata beyond the life of the service
T1	Terms of use are clearly and publicly stated
T2	Sources of funding are clearly and publicly stated



Legend of symbols used in the analysis table below:

+	“complies”: The target resource satisfies the requirement
*	“complies only partially”: The target resource partially satisfies the requirement
WIP	“work in progress”: The target resource is implementing mechanisms to satisfy this requirement
M	“missing”: The target resource currently does not satisfy this requirement
N/V	“needs to be verified”: The annotator needs to double-check or verify the compliance of this requirement
N/A	“not applicable”: The requirement does not apply to the analyzed resource

Zenodo

Overview
Zenodo is a general-purpose repository that enables researchers, scientists, projects and institutions to share, preserve and showcase multidisciplinary research results (data, software, publications, etc). It is founded in the trustworthy CERN data centre, and it is managed, developed and maintained by CERN, although funding comes also from other sources like: EC through OpenAIRE (main partner), SLOAN foundation, and Arcadia. Zenodo hosts more that 2 million records in total, more than 250,000 software records (including all versions) and around 500 TBs of files.

URL: Zenodo

EOSC: https://marketplace.eosc-portal.eu/services/zenodo-bdf48aca-1511-4ab6-8009-9d049dd9c876

Users
The primary users are researchers. The service is open and free for everyone. The use of some functionalities requires an approved account.

Scope
- Geographical scope: international
- Content scope: multi-disciplinary

Existing services/functionalities
- Source code archival: own storage
- Supported identifiers: extrinsic
- Curation of metadata: Manual curation by submitter (updates version)
- Citation formats
- Export formats
- GitHub integration for automatic archival of software

Planned services/functionalities (WIP)
- Integration with SWH with the deposit feature (within the FAIRCORE4EOSC EU project)
- Codemeta.json export
- BibTeX export with specific software type

Adoption
Zenodo is widely used by the global research community. It is a trusted service that saw a huge increase in the last few years. Nowadays, Zenodo receives more than 25 million visits a year.

Documentation: https://about.zenodo.org/
Reference publication
Code repository: https://github.com/zenodo

EOSC SIRS Gap analysis

A1 +	A2 +	A3 +	A4 +	A5 +	A6 WIP	A7 +	A8 N/A	A9 N/A				
R1 *	R2 +	R3 N/A	R4 N/A	R5 N/A								
D1 M	D2 +	D3 *	D4 +	D5 +	D6 WIP	D7 N/A	D8 +	D9 +	D10 +	D11 N/A	`D1 2 N/A	D13 WIP
C1 WIP	C2 +	C3 WIP	C4 N/A	C5 M	C6 M							
O1 +	O2 +	O3 +	O4 +	O5 +	O6 +							

G1 +	G2 M	G3 M										
S1 *	S2 +	S3 +										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	In progress - integration with SWH by archiving software deposit planned for 2025 as part of FAIRCORE4EOSC
Reference	In progress - exposing Intrinsic identifier SWHID on software record planned for 2025 as part of FAIRCORE4EOSC
Describe	Expose and use intrinsic metadata (found in the source code)
	In progress - integration of CodeMeta planned for 2025 as part of FAIRCORE4EOSC
Cite/credit	In progress - Metadata curation beyond the owner
	In progress - extending current BibTex support, planned for 2025 as part of FAIRCORE4EOSC
Openness	-
Governance	Document governance procedures and participation rules
Sustainability	Find persistent funding mechanism to reduce significant dependance in one-off projects
Transparency	-

HAL

Overview
HAL is the scholarly platform for French scholars

URL: https://hal.science/

Users
scholars; researchers, research professors, PhD students.

Institution (research organisation, university, group of universities, grandes écoles and so on) can manage the scientific output of its researchers and research professors.

Scope

- is an open archive where authors can deposit scholarly documents from all academic fields.
- It is both an application, a unique repository, and a shared platform for institutional archives, thematic open archives and theses.
- HAL is a rich corpus of millions of documents, produced in the context of scientific research and higher education. The platform guarantees access and long-term preservation of these documents and offers a range of services that contribute to their enhancement.

Existing services/functionalities

- APIs for : Search resources ; search in the repositories (including the different AureHAL repositories) ; deposit using **API SWORD**, **OAI-PMH** Server, Triplestore)
- a stability of identifiers (URL of deposits in particular),
- a preservation of documents thanks to a partnership with CINES for archiving,
- a scientific quality of documents deposited as well as the details describing them: **all deposited documents are therefore checked before being put on line**,
- a time stamping of deposits guaranteeing the paternity rights of the text deposited,
- **International interconnections with the main archives:**
  - arXiv: automatic transfer of a HAL deposit to arXiv at the request of the depositor,
  - Pubmed
  - EuropePMC,
  - RePec,
  - **OpenAIRE**: visibility of publications associated with a European project.

Other services:

- author identifier and CV,
- browsing statistics and downloads,
- customisable website (publications collection)

	• simple extraction of publications, browsing statistics and downloads…. **Planned services/functionalities (WIP)** • Link with external preprint peer reviews services • • Opening source code
Adoption • The largest research organisations and the majority of French universities have chosen and support HAL, a public, sustainable and responsible infrastructure. • Beyond France, the Knowledge Society IFIP has chosen HAL for its Digital Library	**RefereDocumentation** **APIs:** *https://api.archives-ouvertes.fr/docs* **TripleStore:**(Documentation, SPARQL EndPoint and Download: https://data.hal.science/

EOSC SIRS Gap analysis

A1 +	A2 +	A3 +	A4 +	A5 +	A6 +	A7 +	A8 +	A9 +				
R1 +	R2 +	R3 +	R4 +	R5 +								
D1 +	D2 +	D3 *	D4 +	D5 +	D6 +	D7 N/A	D8 +	D9 +	D10 +	D11 N/A	`D12 N/A	D13 +
C1 +	C2 +	C3 +	C4 +	C5 M	C6 N/A							
O1 +	O2 +	O3 +	O4 WIP	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 WIP	S3 WIP										
T1 *	T2 +											

Gaps identified	Possible Actions
Archive	-
Reference	-
Describe	Improve the feature to linked software deposits and other objects : publications (e.g. articles, …) and data. To better measure the impact of software, perhaps reword some types of links already proposed or add new ones, such as: ● describes / is described by: the article provides a description of the software ● uses / is used by: the article reports work that use the software
Cite/credit	Managing a signatory structure as a collective author to give credit to person and group
Openness	Make the source code open
Governance	-
Sustainability	In progress - Make public the long-term sustainability plan
Transparency	Terms of use exist and are publicly stated but still need to be clarified

eCienciaDatos

Overview
eCienciaDatos is the scholarly platform for researchers of the “madroño” consortium, which includes a group of six Universities in Madrid (UAH, UPM, UCM, UC3M, UNED, URJC)

URL:

https://edatos.consorciomadrono.es/

Users

Users from Consortium

Scope

- Multi disciplinar scope (social sciences, humanities, engineering, biology, mathematics, medicine, etc..)

Existing services/functionalities
Based on Dataverse
Contents are harvested by OpenAIRE
Integrated with OpenAIRE

Planned services/functionalities (WIP)

Adoption
Adopted By scholars of the consortium

Documentation
http://www.consorciomadrono.es/docs/eCienciaDatosTechnicalDescription.pdf

Reference publication

Code repository
Based on Dataverse but not publicly available
See:
https://github.com/IQSS/dataverse

EOSC SIRS Gap analysis

A1 *	A2 +	A3 *	A4 +	A5 +	A6 +	A7 +	A8 +	A9 *				
R1 M	R2 +	R3 M	R4 +	R5 M								

D1 M	D2 +	D3 M	D4 *	D5 +	D6 *	D7 M	D8 *	D9 M	D10 M	D11 N/A	D12 +	D13 +
C1 M	C2 *	C3 +	C4 M	C5 M	C6 M							
O1 *	O2 *	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 +	S3 +										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	Adoption of specific schema for eCienciaDatos software metadata
Reference	Support eCienciaDatos software with their own intrinsic identifier
	Only DOIs from DataCite are supported software bundles, other extrinsic identifiers for software projects must be supported
Describe	Harvest intrinsic metadata of deposited software and store them with the respect to the best practice
	Develop features to allow editing metadata
	Support domain specific ontologies for software metadata

	Develop features to allow metadata supporting relations with other research object and other identifiers
Cite/credit	Develop features to allow software metadata to support contributor role among authors
	Extend existing Bibtex vocabulary to software entries to ease citation of the software artifact (cf: https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/-/blob/master/swentry.org)
	Support the BiblateX software package for typesetting the citation data
	Adoption of tools for plagiarism detection
Openness	An API exists but the documentation must be written and made available
Governance	-
Sustainability	-
Transparency	-

Figshare

Figshare is a repository where users can make all of their research outputs available in a citable, shareable and discoverable manner

URL:

https://figshare.com/

Users
Scholars, PhD Students

Scope
- Data, Softwares, scientific papers

Existing services/functionalities
https://zenodo.org/record/3946720

Planned services/functionalities (WIP)
https://eu-rm.roadmunk.com/publish/50c0cac4ff2d9b46f7c118eb347f7959ffc9f48a

Adoption
Free platform - 0.5m users
Paid services - >100 universities globally. Funders. Publishers.

Documentation
https://knowledge.figshare.com/
API: https://docs.figshare.com/
Reference publication
https://doi.org/10.6084/m9.figshare.4742866.v1
Code repository
Closed github

EOSC SIRS Gap analysis

A1 +	A2 +	A3 +	A4 +	A5 +	A6 M	A7 +	A8 N/A	A9 N/A				
R1 +	R2 +	R3 N/A	R4 N/A	R5 N/A								
D1 M	D2 +	D3 *	D4 *	D5 +	D6 *	D7 +	D8 +	D9 +	D10 +	D11 M	D12 M	D13 +
C1 M	C2 *	C3 +	C4 +	C5 M	C6 N/A							

O1 +	O2 +	O3 +	O4 N/V	O5 +	O6 M							
G1 M	G2 M	G3 M										
S1 +	S2 +	S3 +										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	Figshare must feed universal archive
	Codemeta must be supplied for software metadata
Reference	Necessits to identify the repositories equipped with explicit id
Describe	Develop features to allow editing metadata
Cite/credit	Adopt plagiarism detection tools and support software contributors
Openness	-
Governance	Governances should clearly stated and publicly available
Sustainability	Financial and long term plan of the infrastructure should be available
Transparency	-

RepOD

Overview

RepOD is a scholarly data repository supporting scholars from all disciplines and all parts of the world, but it is *de facto* focused on Polish scholarly community.

It is run by the Interdisciplinary Centre for Mathematical and Computational Modelling at the University of Warsaw.

The deposits are made by individual scholars interested in sharing their data, but there are also institutional collections (dataverses) for individual institutions (for instance a university or a research institute) that are managed by these institutions.

URL:

https://repod.icm.edu.pl/

Users
Researchers wanting to deposit scholarly data..

Scope

- RepOD allows deposition of research data (including scripts) from all disciplines.

Existing services/functionalities

- Storing, archiving, sharing, browsing and searching data and metadata.
- Basic verification of all dataset before publication.
- DOI is provided by the repository.
- Based on Dataverse v. 4.11, but with additional features built on top of it.
- Fixed list of common licences (CC0, CC BY, CC BY SA, Apache, GNU Licences, MIT, ODbL).
- All metadata available on CC0.
- OAI-PMH server.
- APIs for data management.
- Embargo feature (max. 36 months).
- Geospatial metadata (geographic bounding box) visualisation via OpenStreetMaps.
-

Planned services/functionalities (WIP)

- CoreTrustSeal application pending.

Adoption
Who is using RepOD

Documentation
https://repod.icm.edu.pl/info/

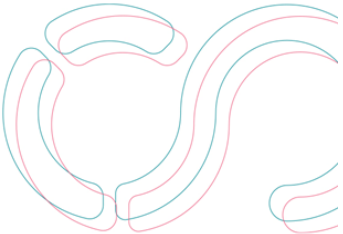
Scholars, PhD Students, Students, all interested stakeholders, including those from outside academia	**Reference publication** **https://repod.icm.edu.pl/info/** **Code repository** https://github.com/CeON/dataverse/

EOSC SIRS Gap analysis

A1 *	A2 +	A3 +	A4 +	A5 +	A6 M	A7 +	A8 N/A	A9 N/A				
R1 M	R2 +	R3 M	R4 M	R5 M								
D1 M	D2 +	D3 *	D4 +	D5 +	D6 M	D7 M	D8 +	D9 +	D10 *	D11 M	`D12 N/A	D13 +
C1 M	C2 M	C3 +	C4 M	C5 M	C6 N/A							
O1 +	O2 +	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 N/A										
S1 +	S2 +	S3 +										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	RepOD should support CodeMeta and feed the Universal Software Archive (Software Heritage)
	Storing metadata versioning
Reference	RepOD must support repositories with intrinsic identifiers like the SWHID
	Publications should reference software extrinsic identifiers

Describe	Adapt existing vocabularies software
	Build ontologies for software
Cite/credit	Adoption of tools for classification of contributor roles for research software Implement bibliographic citation models of software Bibtex already implemented but not specific software entries
Openness	Adopt plagiarism detection tools (using external API)
Governance	-
Sustainability	-
Transparency	-



KU Leuven RDR

Overview
KU Leuven RDR is a scholarly data repository supporting KU Leuven scholars

URL:

https://rdr.kuleuven.be/

Users
Exclusively KU Leuven scholars can deposit data and software on the platform.
Data downloading and reuse is open to everyone.

Scope

- KU Leuven RDR is the data repository supporting KU Leuven scholars in publication of their data and software.
- The repository is open for data and code publication across all research domains practiced at KU Leuven. So, the repository is domain-agnostic.

Existing services/functionalities

Based on Dataverse
Curation/review phase
Integration dashboard for pulling in data from active data systems (Github, Gitlab, iRods, OSF…)
OpenAire provider
EOSC marketplace provider/data source

Planned services/functionalities (WIP)

- Adding resource type 'software' besides the 'dataset' type for publishing code
- Applying for CoreTrustSeal
- Improving input of metadata using external vocabularies
- Investigating Globus integration
- Extending the integration dashboard to include more active data systems (RedCap, OneDrive, SharePoint).
-

Adoption

As expected: KU Leuven researchers publish data on the platform and external (and internal) users download, access and reuse the published data.

The adoption for data publication is still growing as the platform is relatively new and therefore still relatively unknown.

Documentation
https://www.kuleuven.be/rdm/en/rdr/support-guidelines
Reference publication
https://rdr.kuleuven.be/
Code repository
https://rdr.kuleuven.be/

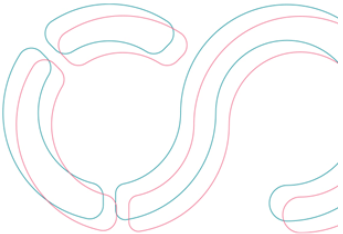
EOSC SIRS Gap analysis

A1	A2	A3	A4	A5	A6	A7	A8	A9				
*	+	+	+	+	M	+	N/A	N/A				

R1 +	R2 +	R3 *	R4 +	R5 *								
D1 N/A	D2 +	D3 +	D4 +	D5 +	D6 M	D7 M	D8 +	D9 +	D10 +	D11 N/A	`D12 N/A	D13 +
C1 +	C2 M	C3 +	C4 +	C5 M	C6 M							
O1 +	O2 +	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 +	S3 +										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	KULeuven RDR should support CodeMeta
	KU Leuven ensures its own archiving and does not consider necessary to feed the universal software archive
Reference	-
	-
Describe	Implementation of the CFF file to harvest intrinsic metadata of the source code
	Storing of the intrinsic metadata
Cite/credit	N/A
Openness	-
Governance	-

Sustainability	-
Transparency	-



Digital CSIC

Overview
DIGITAL.CSIC is the institutional repository of the **Spanish National Research Council**.

URL:

https://digital.csic.es/

Users

scholars

Scope

- **DIGITAL.**CSIC organises, preserves and provides open access to CSIC research outputs.

Existing services/functionalities

- Based on Dspace
- Support software
- Provide guidelines for documenting software

Planned services/functionalities (WIP)

-

Adoption

Documentation
Reference publication

Code repository
Not available

EOSC SIRS Gap analysis

A1 +	A2 +	A3 +	A4 +	A5 *	A6 M	A7 +	A8 N/A	A9 N/A				
R1 +	R2 +	R3 N/A	R4 +	R5 +								
D1 +	D2 +	D3 +	D4 +	D5 +	D6 *	D7 +	D8 +	D9 +	D10 +	D11 +	`D12 N/A	D13 +
C1 +	C2 +	C3 +	C4 +	C5 M	C6 *							

O1 M	O2 +	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 +	S3 +										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	Make software metadata downloadables
	Make software source code archived on Software Heritage
Reference	Necessits to identify the repositories equipped with explicit id
Describe	Many software metadata vocabularies are supported, however CodeMeta is not yet support
Cite/credit	BibteX already implemented, but this one can be extended to software entries (cf: https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/-/blob/master/swentry.org)
	Adopt plagiarism detection tools (using external API)
Openness	Make the CC0 license available for every software metadata
Governance	-
Sustainability	Make public the long term sustainability plan

Transparency	-

Repository Technology

InvenioRDM

Overview

InvenioRDM is an open-source research data management (RDM) platform designed to support the sharing, preservation, and discovery of research data. It provides a comprehensive solution for institutions, organisations, and researchers to effectively manage and publish their research data in a standardised and accessible manner. It offers a user-friendly interface that allows researchers to upload, describe, and organise their datasets, making it easier to document and manage their research outputs throughout the data lifecycle. InvenioRDM is built in collaboration with more than 25 organisations, and it will be the platform behind Zenodo and many other repositories.

URL: https://inveniordm.docs.cern.ch/

Users
The primary users are institutions that wish to provide repository services.

Scope
- Geographical scope: international
- Content scope: multi-disciplinary

Existing services/functionalities
- Source code archival: own storage
- Supported identifiers: extrinsic
- Curation of metadata
- Citation formats
- Export formats
- GitHub integration for automatic archival of software

Planned services/functionalities (WIP)
- Integration with SWH with the deposit feature (within the FAIRCORE4EOSC EU project)
- Codemeta.json export
- BibTeX export with specific software type

Adoption
InvenioRDM is being collaboratively developed by more than 25 global partners. Many instances of InvenioRDM are already operative in several organisations (e.g. http://data.caltech.edu/). Zenodo will be running on InvenioRDM by the end of 2023, and many other repositories will follow.

Documentation: https://inveniordm.docs.cern.ch/
Code repository: https://github.com/inveniosoftware/invenio-app-rdm

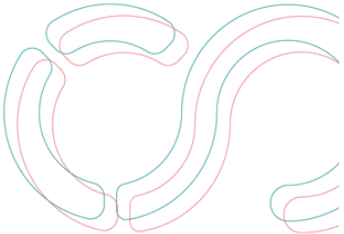
EOSC SIRS Gap analysis

A1	A2	A3	A4	A5	A6	A7	A8	A9				
+	+	+	+	+	WIP	+	N/A	N/A				

R1 WIP	R2 +	R3 N/A	R4 N/A	R5 N/A								
D1 M	D2 +	D3 *	D4 +	D5 +	D6 WIP	D7 N/A	D8 +	D9 +	D10 +	D11 N/A	`D12 N/A	D13 WIP
C1 +	C2 +	C3 WIP	C4 N/A	C5 M	C6 M							
O1 +	O2 +	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 N/A	S3 N/A										
T1 N/A	T2 +											

Gaps identified	**Possible Actions**
Archive	In progress - integration with SWH by archiving software deposit planned for 2025 as part of FAIRCORE4EOSC
Reference	In progress - exposing Intrinsic identifier SWHID on software record planned for 2025 as part of FAIRCORE4EOSC
Describe	Expose and use intrinsic metadata (found in the source code)
	In progress - integration of CodeMeta planned for 2025 as part of FAIRCORE4EOSC
Cite/credit	In progress - Metadata curation beyond the owner
	In progress - extending current BibTex support, planned for 2025 as part of FAIRCORE4EOSC
Openness	-
Governance	-

Sustainability	-
Transparency	-



Publishing platform

EMS Press

Overview EMS Press is the publishing house of the European Mathematical Society **URL:** https://ems.press/	
Users Mathematicians	**Existing services/functionalities**
Scope ● Mathematicians in Europe	**Planned services/functionalities (WIP)**
Adoption	**Documentation** **Reference publication** **Code repository**

EOSC SIRS Gap analysis

(EMS Press team could not be involved in this work)

A1 WIP	A2 +	A3 N/V	A4 N/V	A5 N/A	A6 M	A7 +	A8 N/V	A9 N/V				
R1 +	R2 +	R3 N/A	R4 N/A	R5 N/A								
D1 M	D2 +	D3 M	D4 M	D5 M	D6 WIP	D7 WIP	D8 WIP	D9 WIP	D10 WIP	D11 WIP	`D12 WIP	D13 N/V
C1 WIP	C2 WIP	C3 WIP	C4 WIP	C5 M	C6 WIP							
O1 +	O2 +	O3 WIP	O4 WIP	O5 WIP	O6 WIP							
G1 WIP	G2 WIP	G3 WIP										

S1 WIP	S2 WIP	S3 WIP										
T1 WIP	T2 WIP											

Gaps identified	**Possible Actions**
Archive	-
Reference	-
Describe	-
Cite/credit	-
Openness	-
Governance	-
Sustainability	-
Transparency	-

Dagstuhl Publishing

Overview
Dagstuhl Publishing is a open access publishing house of the Leibniz-Zentrum für Informatik (English: Leibniz Centre for Informatics) Schloss Dagstuhl.
Schloss Dagstuhl's very general mission is to promote the transfer of knowledge between research into informatics and application of informatics, and to operate an international forum and research institute for informatics.

URL: https://www.dagstuhl.de/en/publishing

Users Scholars, researchers	**Existing services/functionalities** **Dagstuhl Publishing** ●
Scope ● Computer Scientist all over the world	**Planned services/functionalities (WIP)** Within the FAIRCORE4EOSC EU project ● Integration of SIRS pillars utilising SWH features (Archive, Deposit). ● CodeMeta.json generator (export). ● CodeMeta.json import. ● CodeMeta.json conversions to other metadata schemes. ● Citation formats export specifying software entry.
Adoption	**Documentation** https://www.dagstuhl.de/en/publishing/series/details/LIPIcs **Reference publication** **Code repository** https://github.com/dagstuhl-publishing/faircore4eosc

EOSC SIRS Gap analysis

A1 WIP	A2 N/A	A3 WIP	A4 WIP	A5 *	A6 +	A7 N/A	A8 N/A	A9 N/A				
R1 *	R2 *	R3 *	R4 WIP	R5 N/A								
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13

WIP	WIP	*	+	+	WIP	*	+	+	WIP	WIP	+	N/A
C1 WIP	C2 +	C3 +	C4 WIP	C5 M	C6 WIP							
O1 +	O2 +	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 +	S3 WIP										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	Ending work in progress of software metadata archiving.
	Ending work in progress for archiving software bundles
	Ending work in progress for archiving public repositories
Reference	Extending the use of extrinsic identifiers
	Ending work in progress to show on document the intrinsic identifiers (SWHID)
Describe	Harvest the intrinsic metadata of the software that has CFF format (or other format) in public repositories (for instance, Github, ..)
	Curate intrinsic metadata (in CFF files)
	Ending work in progress on the development of features to allow editing metadata
	Ending work in progress on the integration of CodeMeta vocabulary

	Provide support to prepare CFF files (intrinsic metadata) based on provided metadata
Cite/credit	Extending the use of CodeMeta conversions to BibTeX and software entries (cf: https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/-/blob/master/swentry.org)
	Ending work in progress for biblatex-software-package
	Provide guidance for authors on correct citation of software
	Adopt plagiarism detection tools
Openness	Make the CC0 license available for every software metadata
	Make the metadata API accessible to non users
	Make the metadata accessible all along the publication process
Governance	-
Sustainability	-
Transparency	-

Episciences

<table>
<tr><td colspan="2">Overview
Publishing platform hosting overlay journals

URL:

https://www.episciences.org/</td></tr>
<tr><td>Users
- Editorial board and reviewers of overlay (scientific) journals hosted on the platform
- Authors submitting (or wishing to submit) an article to overlay journals hosted on the platform</td><td rowspan="2">Existing services/functionalities
- Submission of preprints from open repositories: HAL, arXiv, Zenodo, CWI.
- Peer-reviewing process
- Automatic updating bibliographic information to the host repository: HAL, arXiv
- Metrics available to editorial boards

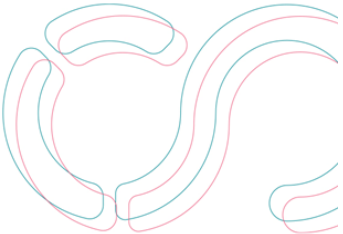
Planned services/functionalities (WIP)
● Editing of extrinsic metadata
● Publishers MUST ensure that software associated with the publication is equipped with proper metadata
● Work on an exit strategy for exiting journals and establish partnership with open repositories which are compliant with TRUST principles</td></tr>
<tr><td>Scope
Overlay journals in the following fields:
- Computer science and applied mathematics
- Mathematics
- Environmental sciences
- Mechanical engineering
- Social sciences and Humanities
● …</td></tr>
<tr><td>Adoption
26 journals running on the platform, 5609 articles published, 11836 users</td><td>Documentation
https://episciences.readthedocs.io/en/latest/
Reference publication
https://hal.science/hal-02148991
Code repository
https://github.com/CCSDForge/episciences</td></tr>
<tr><td colspan="2">EOSC SIRS Gap analysis</td></tr>
</table>

A1 N/A	A2 N/A	A3 WIP	A4 +	A5 +	A6 +	A7 +	A8 +	A9 +				
R1 N/A	R2 +	R3 N/A	R4 +	R5 +								
D1 N/A	D2 +	D3 M	D4 +	D5 +	D6 +	D7 N/A	D8 +	D9 +	D10 +	D11 +	D12 WIP	D13 N/A
C1 N/A	C2 M	C3 M	C4 M	C5 M	C6 N/A							
O1 +	O2 +	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 *	S3 WIP										
T1 +	T2 *											

Gaps identified	**Possible Actions**
Archive	Establish partnership with open repositories which are compliant with TRUST[17] principles
Reference	-
Describe	-
Cite/credit	Implementation of the bibtex software entries to ease citation of the software artifact (cf: https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/-/blob

[17] Lin et al., “The TRUST Principles for Digital Repositories.”

	/master/swentry.org) Support the biblateX software package for typesetting the citation data Provide plagiarism detection tools
Openness	-
Governance	-
Sustainability	Define an exit strategy that could give continuity to the data and metadata beyond the life of the service
Transparency	Make public the sources of funding and the plan for long-term availability of the service



IPOL

Overview
IPOL is an open-science research journal on reproducible algorithms which emphasises the role of mathematics as a source for algorithm design.

URL:

https://www.ipol.im/

Users
Applied mathematicians working on signal analysis and processing algorithms

Scope

- IPOL focuses on reproducible research on signal processing and analysis algorithms, which is the scope of researchers working on the field. Also the industry, as IPOL reveals the current state of the art.

Existing services/functionalities

- IPOL supports publications made of articles + source code. Publication of datasets accepted.
- Open and free platform for the execution of the algorithms
- Strict peer-review of the submitted source code
- Systematic archiving of published source codes in Software Heritage

Planned services/functionalities (WIP)

- Use of Docker containers to ensure long-term reproducibility
- Use of git to live-develop demos

Adoption

- According to IPOL's statistics, the articles and demos of the journal are downloaded and used by the academy as well as the industry. Private companies have negotiated with the authors of IPOL code under the GPL license the possibility of dual-licensing the software to be used in their products.
- IPOL is a well-known open-science and reproducible research journal, specially in the field of image processing.

Documentation
https://www.ipol.im/meta/policy/
https://tools.ipol.im/wiki/ref/software_guidelines/

Reference publication
A. Nicolaï et al. The approach to reproducible research of the Image Processing On Line (IPOL) journal. Informatio, 2022, 27 (1). https://hal.science/hal-04122026

Code repository
https://github.com/ipol-journal/ (all)
https://github.com/ipol-journal/ipolDevel (demo system)
Source code each published methods available in the IPOL's website (ex: https://www.ipol.im/pub/art/2018/236/mlhelPOL.tgz) and also in Software Heritage

	(ex: https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://doi.org/10.5201/ipol.2018.236)

EOSC SIRS Gap analysis												
A1 +	A2 N/A	A3 M	A4 M	A5 +	A6 +	A7 +	A8 +	A9 +				
R1 +	R2 +	R3 +	R4 +	R5 +								
D1 M	D2 +	D3 M	D4 *	D5 *	D6 WIP	D7 *	D8 +	D9 +	D10 +	D11 +	`D12 WIP	D13 N/A
C1 M	C2 M	C3 *	C4 *	C5 M	C6 +							
O1 +	O2 +	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 M	S2 *	S3 +										
T1 +	T2 M											

Gaps identified	**Possible Actions**
Archive	Make metadata for software editable and curated in the publishing process
Reference	-
Describe	Curation of intrinsic metadata
	Support domain specific ontologies for software metadata

Cite/credit	Adoption/Development of tools for classification of contributor roles for research
	Implementation of the Bibtex software entries to ease citation of the software artifact (cf: https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/-/blob/master/swentry.org)
	Support the BiblateX software package for typesetting the citation data
	Adoption of tools for plagiarism detection
Openness	-
Governance	-
Sustainability	There is definitely a need for long term funding as IPOL fully depends on projects to insure the availability of its service
Transparency	Declare all public sources of funding

Aggregator

OpenAIRE

<table>
<tr><td colspan="2">Overview

OpenAIRE is a non-profit legal entity offering networking services and technical services to favour the implementation and adoption of Open Science practices in Europe and beyond. One of the core technical services we offer is the OpenAIRE Research Graph an open, transparent, metadata collection bringing in all scholarly communication sources world wide. We collect metadata from around 12,000 sources (Crossref, DataCite, Unpaywall, MAG, ORCID, GRID/ROR, preprints, institutional repositories from OpenDOAR, etc.), organise scientific results in publications, datasets, and software, and interlink them with funders, projects, organisations (and the data sources from which we collect them). The Graph counts 110 Pubs, 7 Mi datasets, 200K software, 30 funders, 3.5 million projects, and around 1Bi relationships between such objects. OpenAIRE is one of the pillars of the European Open Science Cloud.

URL: https://explore.openaire.eu/ and https://openaire.eu/

EOSC: https://marketplace.eosc-portal.eu/providers/eosc.openaire/details</td></tr>
<tr><td>Users
The main users of OpenAIRE Explore are researchers, scientists, and anyone seeking open access research publications and datasets.</td><td rowspan="2">Existing services/functionalities
• Search for open access research publications and datasets.
• Discover related research based on keywords, funders, projects, authors, or topics.
• Analyze and visualize trends in research outputs.
• Integrate with other research infrastructures and platforms.
• Ensure compliance with open access policies and mandates.

Planned services/functionalities (WIP)</td></tr>
<tr><td>Scope
• Geographical scope: international
• Content scope:</td></tr>
<tr><td>Adoption
OpenAIRE's services, including OpenAIRE Explore, have played a crucial role in promoting open access research and facilitating the discovery and accessibility of scholarly outputs.</td><td>Documentation:
https://www.openaire.eu/guides
Reference publication
Code repository</td></tr>
<tr><td colspan="2">EOSC SIRS Gap analysis</td></tr>
</table>

A1 N/A	A2 N/A	A3 N/A	A4 N/A	A5 M	A6 N/A	A7 N/A	A8 N/A	A8 N/A	A9 N/A			
R1 M	R2 +	R3 M	R4 M	R5 M								
D1 M	D2 +	D3 +	D4 M	D5 +	D6 M	D7 N/A	D8 +	D9 +	D10 +	D11 N/A	`D12 N/A	D13 N/A
C1 *	C2 M	C3 +	C4 M	C5 N/A	C6 N/A							
O1 +	O2 +	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 +	S3 +										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	-
Reference	Expose software artifacts along with intrinsic identifiers if available
Describe	Support CodeMeta for Research Software entries
Cite/credit	Complete support to software-related roles in metadata, and expand bibliographic citation data model to adapt to software specific needs
Openness	-
Governance	-
Sustainability	-

Transparency	-

swMATH

<table>
<tr><td colspan="2">The zbMATH Open Software Search is based on results of the swMATH Project, which is currently conducted by FIZ Karlsruhe and Zuse Institute Berlin (ZIB) within the scope of Forschungscampus Modal funded by the BMBF.

URL:

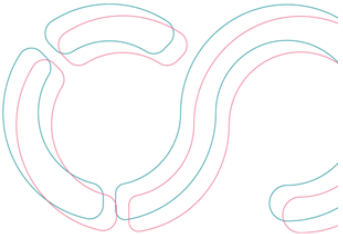
https://swmath.org</td></tr>
<tr><td>Users
The primary users are mathematicians. The service is open and free for everyone. Mathematicians must submit an article and the linked associated software to have its metadata properly indexed, stored and exposed through its interface</td><td rowspan="2">Existing services/functionalities
Browsing mathematical software
Supported identifiers: swMATH identifiers
Curation of metadata: Manual curation by swMATH curator
Linked scholar articles with zbMATH identifiers
Planned services/functionalities (WIP)
Integration with SWH with the metadata deposit feature (within the FAIRCORE4EOSC EU project)
swMATH will ensure archiving source code linked to its metadata
Codemeta and DataCite will be available for export</td></tr>
<tr><td>Scope
Geographical scope: international
Content scope: Mathematics</td></tr>
<tr><td>Adoption

swMATH is widely adopted in the mathematical community</td><td>Documentation
https://zbmath.org/about/
Reference publication
https://euromathsoc.org/magazine/articles/118
Code repository
https://github.com/MaRDI4NFDI/swMATH4EOSC</td></tr>
<tr><td colspan="2">EOSC SIRS Gap analysis</td></tr>
</table>

A1 WIP	A2 N/A	A3 +	A4 +	A5 WIP	A6 N/A	A7 N/A	A8 N/A	A9 N/A				
R1 +	R2 WIP	R3 M	R4 M	R5 M								
D1 M	D2 +	D3 +	D4 M	D5 M	D6 +	D7 +	D8 +	D9 +	D10 +	D11 N/A	D12 N/A	D13 N/A
C1 M	C2 N/A	C3 M	C4 M	C5 N/A	C6 N/A							
O1 M	O2 M	O3 +	O4 WIP	O5 WIP	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 +	S3 M										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	swMATH must support CodeMeta
	swMATH must achieve the ongoing work with FairCore4EOSC to enrich SWH with swMATH metadata
Reference	Harvest the intrinsic identifier of the software through the SWH API and expose it in the interface
	End the work in progress on the support of extrinsic identifier for software bundles
	Support extrinsic identifier for software projects
Describe	Software intrinsic metadata as found in CFF file must enrich the swMATH metadata
Cite/credit	Metadata editor should allow contributors classification

	Support the Bibtex format with specific software entries to ease citation in publications
Openness	Make the CC0 license available for every software metadata
	Make the software metadata available API through an API
Governance	-
Sustainability	Build an exit strategy to ensure continuity for metadata
Transparency	-



Research Software Discovery

Overview

The research software directory is an online registry for sharing research software, making it more visible and discoverable, showing its academic and societal impact, and promoting software citation. Originally developed by the Netherlands eScience Center in 2018 to showcase its research software output, it is currently a joint effort with the Helmholtz Association and Imperial College London. The RSD is available as a free online service for researchers, research software engineers and research organisations across all domains. Organisations can also host their own (customised) RSD if they desire to do so.

URL: https://research-software-directory.org

Documentation: https://research-software-directory.github.io/documentation/

Helmholtz Association deployment: https://helmholtz.software

Users

The service is open to use for researchers, research software engineers and research organisations across all domains, but does require an account. Individuals may sign-up using ORCID as authentication. Dutch research institutes may connect thru SURFConext, allowing their employees to sign in using their institute credentials. Similarly Helmholtz AAI is used for Helmholtz institutes. Additional authentication mechanisms can be added upon request.

Scope

- Geographical scope: International
- Content scope: All domains

Existing services/functionalities

- Indexing research software and providing search functionality to increase research software findability.
- Highlighting the impact of research software by linking it to other research outputs and activities, such as projects, publications, presentations, datasets, workshops, teams, etc.
- Encouraging software citation by providing citation metadata for research software.
- Exporting metadata to other services such as OpenAire

Planned services/functionalities (WIP)

Providing metrics on software impact by collecting information on software reuse, citation, etc. Both for individual software items, as well as aggregated metrics per organization.

Adoption

- The main directory is currently used by 8 Dutch research institutes, and several

Documentation:

https://research-software-directory.github.io/documentation/

international ones. The Helmholtz Association has its own deployment used by 15 institutes. Imperial College London is currently setting up its own deployment. • The tool is open source and can be potentially adopted by any institution. The main directory is currently free of charge.	**Reference publication** **Code repository,** https://github.com/research-software-directory/RSD-as-a-service

EOSC SIRS Gap analysis

A1 +	A2 N/A	A3 +	A4 WIP	A5 n/a	A6 WIP	A7 N/A	A8 N/A	A9 N/A				
R1 M	R2 +	R3 M	R4 +	R5 M								
D1 M	D2 +	D3 WIP	D4 +	D5 +	D6 +	D7 +	D8 +	D9 WIP	D10 WIP	D11 N/A	`D12 +	D13 N/A
C1 *	C2 *	C3 +	C4 N/A	C5 M	C6 N/A							
O1 +	O2 +	O3 +	O4 +	O5 +	O6 +							
G1 WIP	G2 WIP	G3 WIP										
S1 WIP	S2 WIP	S3 WIP										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	Trigger the archival of software source code and their associated metadata
Reference	Expose the SWHID of software source code with the help of of the SWH API

	References to other software must be supported with proper extrinsic identifiers
Describe	Intrinsic metadata must be harvested from repositories like from the source code itself
Cite/credit	Introduce contributor role classification.
	BibTeX is already supported, however it must be extended to specific entries for software like the ones proposed in : https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/-/blob/master/swentry.org
	Adoption of tools for plagiarism detection
Openness	-
Governance	Define a clear governance structure and make it publicly available
Sustainability	Define a clear long term technical and financial sustainability plan
	An exit strategy must be elaborated to ensure the long term preservation of the data and the metadata beyond the life of the service
Transparency	-

bio.tools

Overview
Essential scientific and technical information about software tools, databases and services for bioinformatics and the life sciences.

URL:

EOSC: [market-place record link]

Users
The primary users are researchers. The service is open and free for everyone. The use of some functionalities requires an approved account (for example for submitting metadata).

Scope
- Geographical scope: international
- Content scope: Mathematics

Existing services/functionalities
- Source code archival: using SWH webhook
- Supported identifiers: extrinsic
- Curation of metadata: Manual curation by curator (updates version)
- Citation formats: [to complete]
- Export formats: [to complete]

Planned services/functionalities (WIP)
- Integration with SWH with the metadata deposit feature (within the FAIRCORE4EOSC EU project)
- Codemeta.json export
- BibTeX export with specific software type

Adoption

Documentation
Reference publication
Code repository

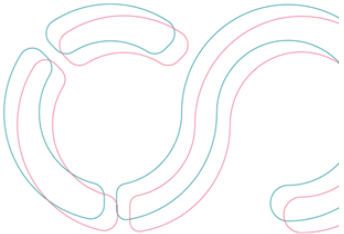
EOSC SIRS Gap analysis

(bio.tools team could not be involved in this work)

A1 N/A	A2 N/A	A3 N/A	A4 N/A	A5 N/A	A6 N/A	A7 N/A	A8 N/A	A9 N/A				
R1 +	R2 +	R3 N/V	R4 N/V	R5 N/V								
D1 N/A	D2 +	D3 +	D4 +	D5 +	D6 -	D7 N/A	D8 +	D9 +	D10 +	D11 N/A	`D12 N/A	D13 N/A

C1 *	C2 M	C3 M	C4 M	C5 M	C6 M							
O1 +	O2 M	O3 +	O4 +	O5 +	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 +	S3 +										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	-
	-
Reference	-
	-
Describe	-
	-
Cite/credit	-
Openness	-
Governance	-
Sustainability	-
Transparency	-



Aggregator Technology

Geant Software Catalogue

Overview
GÉANT Software Catalogue is a catalogue of software projects and teams, established in order to support the GÉANT software engineering community through exploration of distributed project resources (code repositories, issue trackers, etc.) and collecting up-to-date information about state and progress of GÉANT software development efforts. This is a software that can be instantiated and used free of charge. It's not a public service.

GEANT Software Catalogue (deployed on the GEANT infrastructure):

https://sc.geant.org/

GEANT Software Catalogue (the software, that can be deployed by any institutions): https://bitbucket.software.geant.org/projects/SC/repos/softwarecataloguesuite/browse

Users
The application is only available to users of specific organization, institution, or company. For example, GÉANT project participants can access installation of SC that requires authentication and is restricted to GEANT users only but is free of charge.
The primary users of the GEANT deployment are researchers developing software for the GEANT network ecosystem as well as quality engineers auditing the developed software.

Scope
- Geographical scope: International
- Content scope: All domains

Existing services/functionalities
- Indexing software projects
- Building a big-picture of a software project that is spread over multiple locations (code repositories, issue trackers, etc.)
- Builds a catalogue of indexed software projects, providing reports and searching functionality

Planned services/functionalities (WIP)
Make the catalogue hybrid in terms of content visibility to public Internet. The catalogue will get an option to become fully open. That would mean ability to access its parts without authentications, such as: searching, reporting and software projects records.

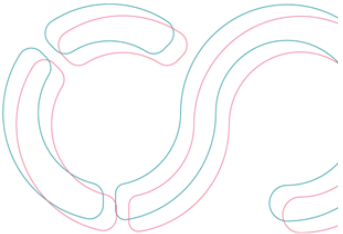
Adoption
- GEANT Project has adopted SC as an internal tool. Access is restricted to authenticated GEANT users.
- The tool is free of charge and can be potentially adopted by any institution.

Documentation:
https://wiki.geant.org/display/GSD/Software+Catalogue,
Reference publication

							Code repository, https://bitbucket.software.geant.org/projects/SC/repos/softwarecataloguesuite/browse					
EOSC SIRS Gap analysis												
A1 N/A	A2 N/A	A3 N/A	A4 N/A	A5 N/A	A6 N/A	A7 N/A	A8 N/A	A9 N/A				
R1 M	R2 +	R3 N/A	R4 +	R5 +								
D1 M	D2 +	D3 M	D4 +	D5 +	D6 M	D7 +	D8 +	D9 M	D10 M	D11 M	`D12 +	D13 ?
C1 *	C2 M	C3 M	C4 M	C5 N/A	C6 *							
O1 M	O2 WIP	O3 WIP	O4 WIP	O5 M	O6 +							
G1 +	G2 +	G3 +										
S1 +	S2 +	S3 N/A										
T1 +	T2 +											

Gaps identified	**Possible Actions**
Archive	-
Reference	Introduce automatic discovery of “whether a code repository” is already available on archive.softwareheritage.org and/or other services where extrinsic identifiers are supported.

Describe	Develop features to allow discovery of existing metadata such as CodeMeta, SPDX, etc.
Cite/credit	Introduce citation data discovery (e.g. existence of CFF files)
Openness	Introduce open web API, use open data formats where possible
Governance	-
Sustainability	-
Transparency	-



10. Appendix A.2

This section introduces the two main vocabularies of the SIRS Gap Analysis.

CodeMeta

<table>
<tr><td colspan="2">Overview
The CodeMeta initiative strives to promote the citation and reuse of software authored for scientific research. It is a vocabulary extending the schema.org SoftwareSourceCode class and a collection of crosswalks between existing software vocabularies and CodeMeta. Using a codemeta.json file in the code repository enables the transfer of software metadata between infrastructures and assists in the archiving, indexing and distribution of software.

URL:

https://codemeta.github.io/</td></tr>
<tr><td>Users
Users are software developers who want to have their software properly cited in scientific publications by curating their metadata.</td><td rowspan="2">Existing services/functionalities
● Supported identifiers: intrinsic and extrinsic
● The CodeMeta Generator
● Web semantic capabilities

Planned services/functionalities (WIP)
● Machine actionable crosswalks</td></tr>
<tr><td>Scope

This vocabulary supports the edition of the metadata describing software.</td></tr>
<tr><td>Adoption
CodeMeta is supported by Software Heritage, Zenodo, HAL, Dagstuhl Publishing, IPOL, swMATH, Research Software Discovery and technologies like Géant and InvenioRDM.</td><td>Documentation
https://codemeta.github.io/user-guide/
Reference publication
https://peerj.com/articles/cs-1023/
Code repository
https://github.com/codemeta/codemeta</td></tr>
</table>

BioSchemas

Bioschemas is an extension of Schema.org providing types for Life Sciences and profiles. A profile is a recommendation built on top of a type on how to use properties (minimum, recommended, optional), cardinality (one, many) and link to other controlled vocabularies. Bioschemas is a community-based collaborative project, aiming to make Life sciences resources more findables. It also supports a number of domain-agnostic research-related profiles such as Datasets or TrainingMaterial.

URL:

https://bioschemas.org/

Users
Primary users are researchers in Life Sciences, with some support for researchers in general.

Scope
This vocabulary supports the representation of software metadata in research.

Existing services/functionalities

- Controlled vocabulary including support for life sciences types (e.g., gene, protein) and research in general (e.g., datasets, articles, software)
- Supported identifiers: extrinsic/intrinsic
- Compatible with CodeMeta and Schema.org

Planned services/functionalities (WIP)

- Metadata generator (current version to be replaced)
- Metadata validation (initial support via FAIR-Checker, will be further developed)
- More domain-tailored tutorials
- Better integration with https://schemas.science
- Types and profiles supporting Software Management Plans and the Software Management Wizard and the Research Data Management Organiser (RDMO)
- Types and profiles supporting Machine Learning life cycle

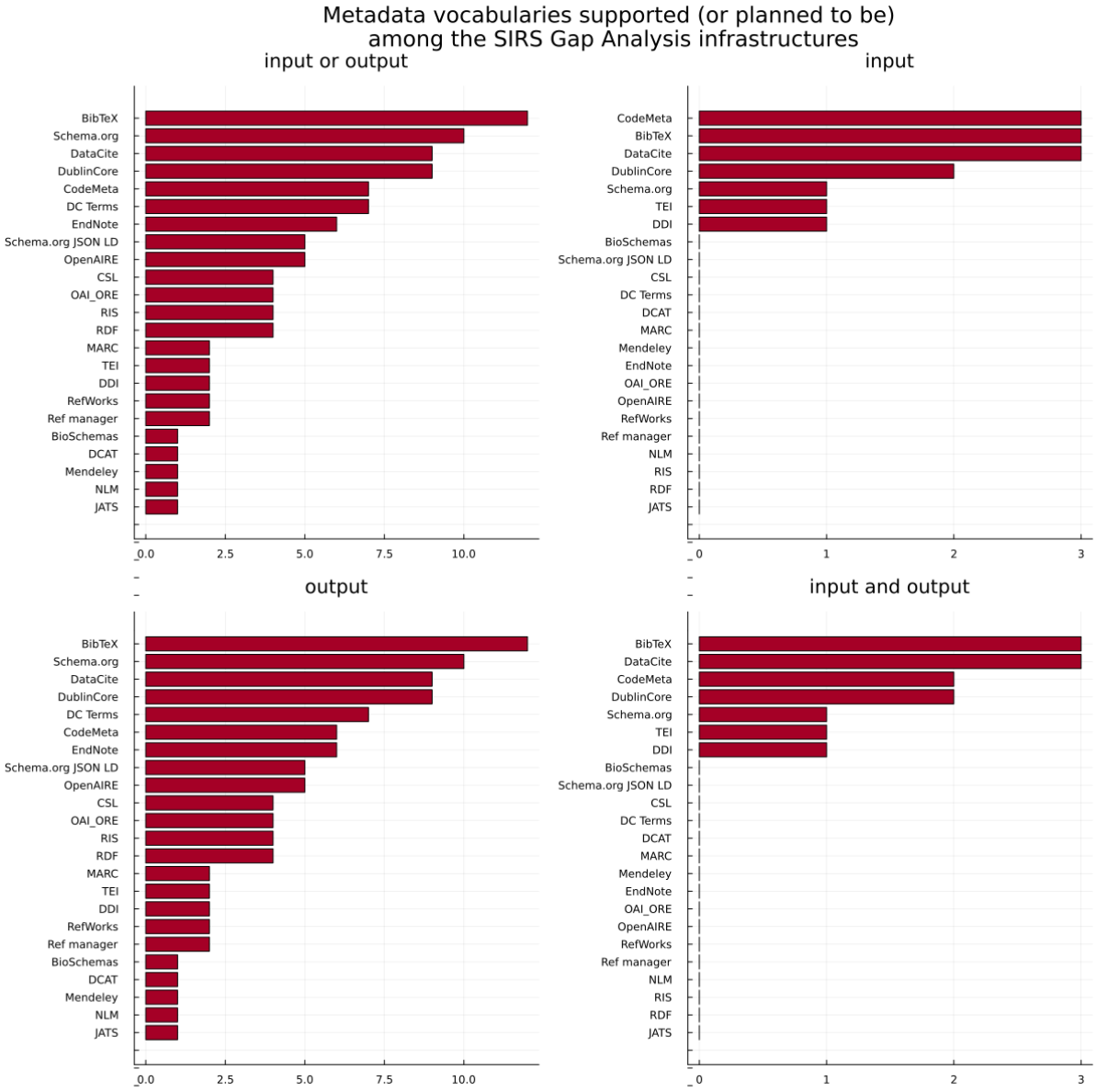
Adoption
Researchers from the Life Sciences domain (and researchers in general, e.g., some NFDI in Germany) use Bioschemas profiles. For the software case, it is supported by bio.tools and the Software Management Wizard. Also supported by TeSS (training materials), RO-Crates (datasets, workflows) and the Workflow Hub (workflows). Implemented in about 100 websites (see list of live deploys)

Documentation
https://bioschemas.org/tutorials/
Reference publication
https://bioschemas.org/about/publications
Code repository
https://github.com/BioSchemas

11. Appendix B.1

Infrastructures	Vocabularies adopted/planned to be adopted and recommended in the SIRS report	
	CodeMeta	BioSchemas
Zenodo	Yes	No
HAL	Yes	No
e-cienciaDatos	Yes	No
ArXiv	No	No
FigShare	No	No
RepOD	No	No
KU Leuven RDR	No	No
Digital CSIC	No	No
InvenioRDM	Yes	No
European Mathematical Society Press (EMS Press)	-	-
Dagstuh Publishingl	Yes	No
EpisciencesDagstuhl Publishing	No	No
Image Processing On Line (IPOL)	No	No
OpenAire	No	No
swMATH	Yes	No
bio.tools	No	Yes
GÉANT Software Catalogue	Yes	No
esciencecenter.nl/research-software-directory/	Yes	No

12. Appendix B.2



***Input**: stands for the possibility to import metadata*
***Output**: stands for the possibility to download metadata*