FAIR 4 Research Software (FAIR4RS) atelier

RDA France

Morane Gruenpeter, Inria, Software Heritage

research data sharing without barriers
rd-alliance.org

November, 2021

Housekeeping

- These slides https://tinyurl.com/RDA-France-FAIR4RS-slides
- Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

- Meeting etiquette RDA Code of Conduct
  - Participants enter in listen only mode
  - Add your questions in the in the chat on in the notes
  - Answer questions and add resources to the collaborative notes
  - Please be aware that the session is being recorded and will be made publicly available

Morane Gruenpeter

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Software engineer and metadata specialist

Timeline:

- 2008-2011 B.A in **Musique** (Harpist)
- 2012-2015 Licence (B.SC) in **Computer Science** @CNAM
- 2015-2017 Master in **Software Engineering** (R&D) @UPMC
- 2017 Internship **Software Heritage** (SWH)
- 2018-2019 European project EU2020 **CROSSMINER**(on SWH team)
- 2020-2022 European project **FAIRsFAIR** (on task T2.4 FAIR services and software)

FAIRSFAIR
Fostering Fair Data Practices in Europe

Working groups for Open Science and digital preservation

- the Research Data Alliance's **Software Source Code** Interest Group (SSC IG),
- the FORCE11's **Software Citation** Implementation Working Group (SCI WG),
- **Chair** of the joint RDA, ReSA & FORCE11 **FAIR for Research Software** Working Group (FAIR4RS WG)
- WikiData for **Digital Preservation** initiative (WikiDigi).

FORCE11
The Future of Research Communications and e-Scholarship

<ReSA>
Research Software Alliance

RDA
RESEARCH DATA ALLIANCE

Welcome and introduction

These slides https://tinyurl.com/RDA-France-FAIR4RS-slides

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

Today’s goals

- Introduction
  - Software in Research
  - Software in a FAIR ecosystem
  - ice-breaker
- Present the FAIR4RS WG activities
- Atelier
  - Review FAIR principles for research software
  - Break (5-10 min’)
  - Clarify the principles and the identified challenges
- Conclusion

What is software?

Software as a concept

- **project** or entity
- the **community** around the project
- the software **idea** / algorithms / solutions

https://www.reddit.com/r/ProgrammerHumor/comments/70fuamp/programming_is_magic/

Software artifact

- **source code** form
  - for each version and revision/commit
- **binaries/executables** produced (for different environments)

FORCE11 The Future of Research Communications and e-Scholarship

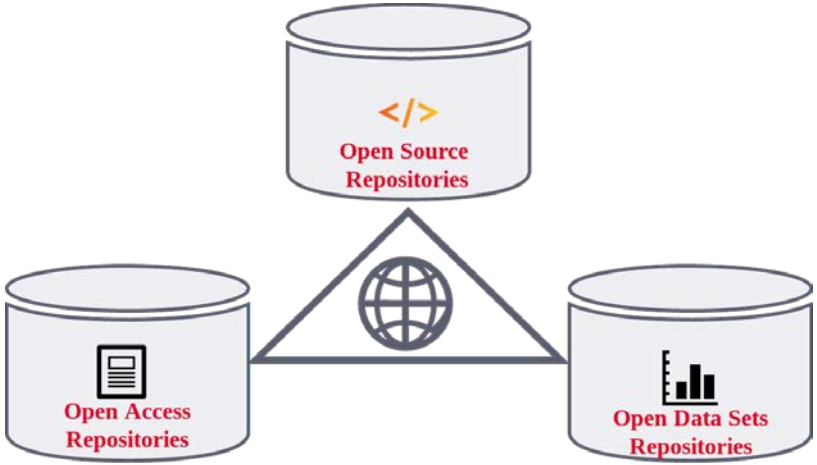
<ReSA> Research Software Alliance

RDA RESEARCH DATA ALLIANCE

Software in Research: A pillar of Open Science

Multiple facets, it can be seen as:

- a **tool**
- a research **outcome** or result
- **the object** of research



Three pillars of Open Science
Gruenpeter, Software Heritage CC-By 4.0 2019

Why are we here? A plurality of needs

Researchers

- **archive and reference** software used and created in articles
- **find** useful software
- **get credit** for developed software
- **verify/reproduce/improve** results

Laboratories/teams

- **track** software contributions
- **produce** reports
- **maintain** web page

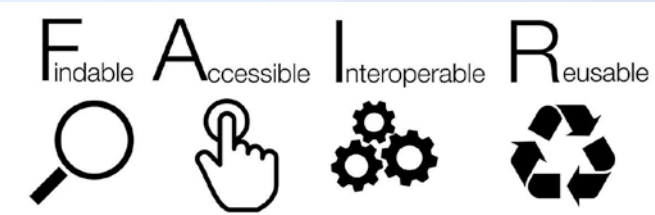
Research Organization

know its **software assets** for:

- technology transfer,
- impact metrics,
- strategy

Software development communities
&
Research Software communities

Building bridges between communities



research data sharing without barriers
rd-alliance.org

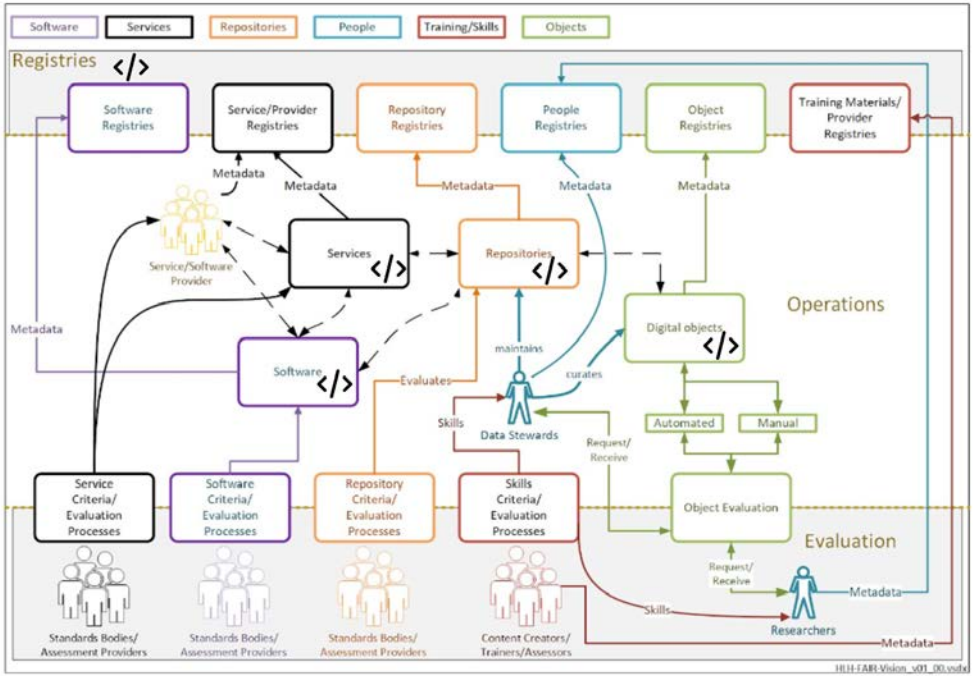
FORCE11
The Future of Research Communications and e-Scholarship

<ReSA>
Research Software Alliance

RDA
RESEARCH DATA ALLIANCE

Software in the FAIR ecosystem

FAIRSFAIR
Fostering Fair Data Practices in Europe



*"Central to the realisation of FAIR are **FAIR Digital Objects**, which may represent data, software or other research resources. These digital objects must be accompanied by persistent identifiers, metadata and contextual documentation to enable discovery, citation and reuse. Data should also be accompanied by the code used to process and analyse the data."*

Rec. 16: *Apply FAIR broadly: "FAIR should be applied broadly to all objects (including metadata, identifiers, software and DMPs) that are essential to the practice of research, and should inform metrics relating directly to these objects."*

Turning FAIR into reality (2018)

Ecosystem components, to highlight the software roles in the Ecosystem, the symbol </> was added (Original diagram 3 from L'Hours & Von Stein, 2020)

Motivation - *Software is not just another type of data*

- FAIR Principles, are intended to apply to **all digital objects** (Wilkinson et al. 2016)
- We focus on the **adaptation and adoption** of the FAIR principles to research software

Recommendation n°5 :

*Recognise that FAIR guidelines will require **translation for other digital objects** and support such efforts.*

2020: 'Six Recommendations for Implementation of FAIR Practice'

(FAIR Practice Task Force EOSC, 2020)

Recommendation n° 2 :

*Make sure **the specific nature of software** is recognized and not considered as "just data" particularly in the context of discussion about the notion of FAIR data.*

2019: the **Opportunity Note** by the French national Committee for Open Science's Free Software and Open Source Project Group (Clément-Fontaine, 2019)

FORCE11 The Future of Research Communications and e-Scholarship

<ReSA> Research Software Alliance

RDA RESEARCH DATA ALLIANCE

FAIRsFAIR report

10.5281/zenodo.4095092

- **Literature review** on the application of FAIR principles to research software
- **State-of-the-art overview** of current solutions, challenges and practices in research software
- **10 recommendations** *for the creation of* FAIR guiding principles for research software

October 16, 2020

Project milestone | Open Access

M2.15 Assessment report on 'FAIRness of software'

Gruenpeter, Morane; Di Cosmo, Roberto; Koers, Hylke; Herterich, Patricia; Hooft, Rob; Parland-von Essen, Jessica; Tana, Jonas; Aalto, Tero; Jones, Sarah

Software has an important place in academia and as such it has an important place in the FAIR ecosystem. Software can be used throughout the research process; however it can also be an outcome of the research process. Distinguishing between these different roles is essential for any assessment of the `FAIRness of software'.

This is the first milestone of the FAIRsFAIR project focused specifically on software as a digital object. In this report we discuss the state-of-the-art of software in the scholarly ecosystem in general and in the FAIR literature in particular. We identify the challenges of different stakeholders when it comes to finding and reusing software. Furthermore, we present an analysis of nine resources that call for the recognition of software in academia and that present guidelines or recommendations to improve its status - either by becoming more FAIR or by improving the curation of software in general. With this analysis we demonstrate to what extent each of the FAIR principles is seen as relevant, achievable and measurable; and in what sense it benefits software artifacts. Finally, we present 10 high-level recommendations for organizations that seek to define FAIR principles or other requirements for research software in the scholarly domain.

Feedback and suggestions will be most welcome as comments on the public Google Doc version of this report
https://docs.google.com/document/d/1yvdLSP6oH3XozVy4CJtThzGNHkseCBdvmxfruDYLB6Q/edit?usp=sharing

Preview

Page : 1 of 63 50%

FAIRSFAIR

M2.15 Assessment report on 'FAIRness of software'

[watch webinar]

research data sharing without barriers
rd-alliance.org

From the FAIRsFAIR report

Keep in mind:

1. Any new principle may lead to **extra requirements enforced on researchers**,
2. Researchers are already facing **significant challenges** when developing or maintaining software, which is **a complex and living object**,
3. **Clear and immediate benefits** should be offered to the researcher.

Ice breaker

Are the FAIR principles (Wilkinson et al.) relevant to Software ?

Les principes FAIR (Wilkinson et al.) sont-ils pertinents pour les logiciels ?

Figure; FAIR in a nutshell. Image: ARDC 2018 - CC-BY 4.0.

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

FAIR4RS WG

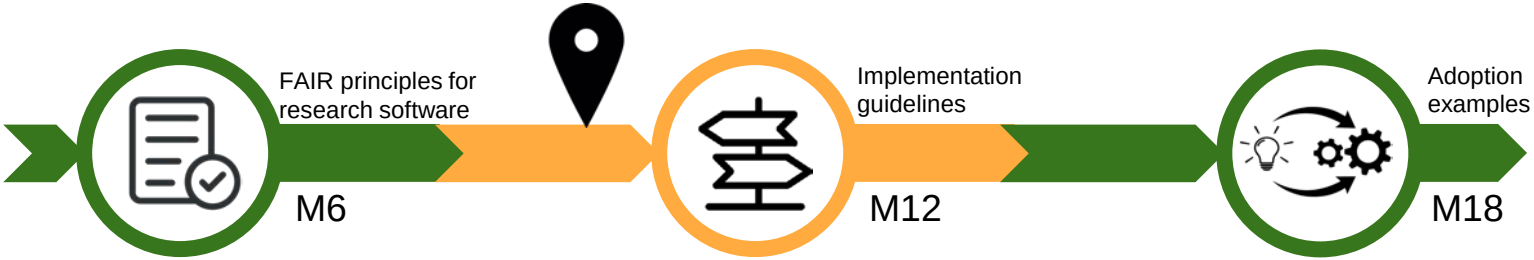
These slides https://tinyurl.com/RDA-France-FAIR4RS-slides

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

Introduction #FAIR4RS

Endorsed by RDA Sep 2020

- A joint **RDA** Working Group, **FORCE11** Working Group, and Research Software Alliance (**ReSA**) Taskforce.
- Coordinating of a range of existing community-led discussions on:
  - How to define and effectively apply FAIR principles to research software,
  - How to achieve adoption of these principles.



https://www.rd-alliance.org/group/fair-4-research-software-fair4rs-wg/case-statement/fair-research-software-wg-case-statement

FORCE11 The Future of Research Communications and e-Scholarship
<ReSA> Research Software Alliance
RDA RESEARCH DATA ALLIANCE

FAIR4RS subgroup activity and outputs

- A fresh look at FAIR for Research Software examined the FAIR principles in the context of research software from scratch, not based on pre-existing work. **Lead: Daniel S. Katz**

- FAIR work in other contexts examined efforts to apply FAIR principles to different forms including workflows, notebooks and training material, to provide insights for the definition and implementation of FAIR principles for research software. **Lead: Michelle Barker**

- Defining Research Software: a controversial discussion reviews existing definitions of research software in order to provide the overall context of the subgroup outputs. **Lead: Morane Gruenpeter**

- Review of new research related to FAIR Software reviewed new research around FAIR software that has come out since the release of the *Towards FAIR principles for research software* (Lamprecht et al., 2019). **Lead: Neil Chue Hong**

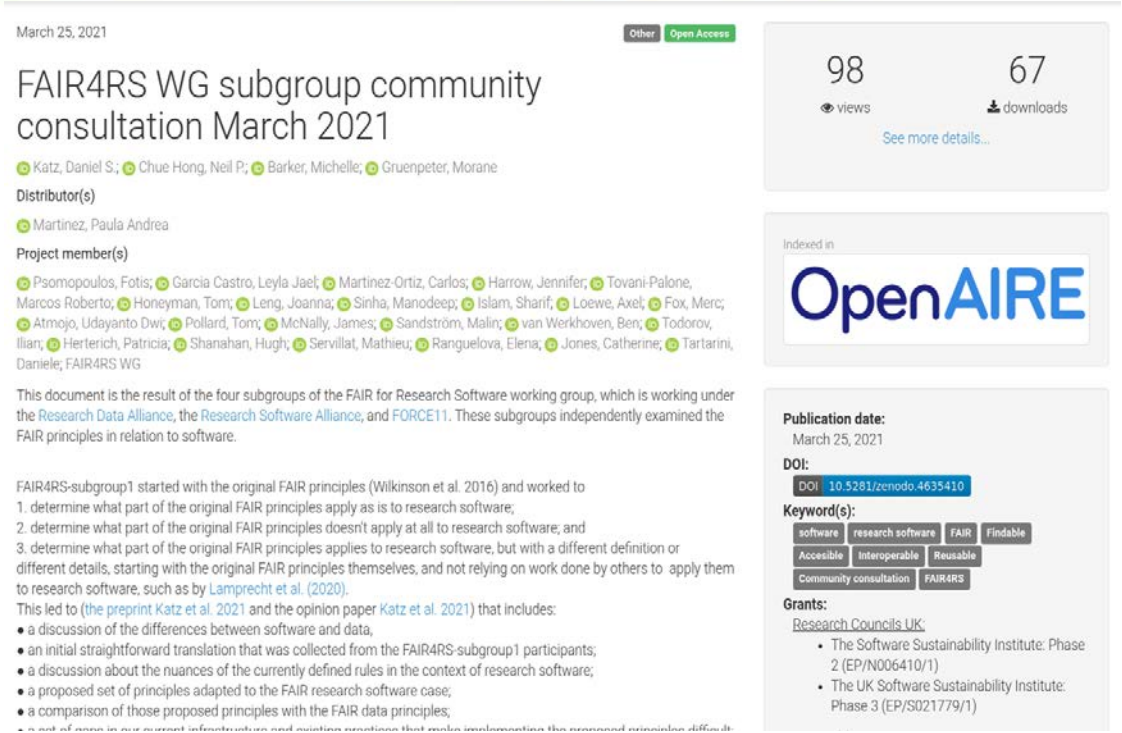
First FAIR4RS community consultation

http://doi.org/10.5281/zenodo.4635410

First community consultation to get feedback on findings of subgroups, and questions around scope of the draft FAIR4RS principles

- **24 February - 10 March 2021**
- 215 comments from 19 named contributors (other than the SC) + other anonymous contributors.

Used as main input for FAIR4RS drafting sprints, with questions and clarifications raised by the community discussed by the drafting team to determining the intent of the principles



March 25, 2021

Other | Open Access

FAIR4RS WG subgroup community consultation March 2021

Katz, Daniel S.; Chue Hong, Neil P.; Barker, Michelle; Gruenpeter, Morane

Distributor(s)

Martinez, Paula Andrea

Project member(s)

Psomopoulos, Fotis; Garcia Castro, Leyla Jael; Martinez-Ortiz, Carlos; Harrow, Jennifer; Tovani-Palone, Marcos Roberto; Honeyman, Tom; Leng, Joanna; Sinha, Manodeep; Islam, Sharif; Loewe, Axel; Fox, Merc; Atmojo, Udayanto Dwi; Pollard, Tom; McNally, James; Sandström, Malin; van Werkhoven, Ben; Todorov, Ilian; Herterich, Patricia; Shanahan, Hugh; Servillat, Mathieu; Ranguelova, Elena; Jones, Catherine; Tartarini, Daniele; FAIR4RS WG

This document is the result of the four subgroups of the FAIR for Research Software working group, which is working under the Research Data Alliance, the Research Software Alliance, and FORCE11. These subgroups independently examined the FAIR principles in relation to software.

FAIR4RS-subgroup1 started with the original FAIR principles (Wilkinson et al. 2016) and worked to
1. determine what part of the original FAIR principles apply as is to research software;
2. determine what part of the original FAIR principles doesn't apply at all to research software; and
3. determine what part of the original FAIR principles applies to research software, but with a different definition or different details, starting with the original FAIR principles themselves, and not relying on work done by others to apply them to research software, such as by Lamprecht et al. (2020).

This led to (the preprint Katz et al. 2021 and the opinion paper Katz et al. 2021) that includes:
- a discussion of the differences between software and data,
- an initial straightforward translation that was collected from the FAIR4RS-subgroup1 participants;
- a discussion about the nuances of the currently defined rules in the context of research software;
- a proposed set of principles adapted to the FAIR research software case;
- a comparison of those proposed principles with the FAIR data principles;

98 views | 67 downloads

See more details...

Indexed in OpenAIRE

Publication date: March 25, 2021

DOI: 10.5281/zenodo.4635410

Keyword(s): software | research software | FAIR | Findable | Accesible | Interoperable | Reusable | Community consultation | FAIR4RS

Grants:

Research Councils UK:
- The Software Sustainability Institute: Phase 2 (EP/N006410/1)
- The UK Software Sustainability Institute: Phase 3 (EP/S021779/1)

research data sharing without barriers
rd-alliance.org

FORCE11 The Future of Research Communications and e-Scholarship

ReSA Research Software Alliance

RDA RESEARCH DATA ALLIANCE

Sprints - Drafting the principles

- small task force
- five meetings in April 2021
- Discussed:
  - feedback from the first consultation
  - Issues from the subgroup work

“maximize the added-value gained by contemporary, formal scholarly digital publishing”

“ensure transparency, reproducibility, and reusability”

The foundational principles of Findable, Accessible, Interoperable, and Reproducible may need to be reinterpreted to ensure that they are applicable to software.

Key challenges:

- Finding the balance between general, more abstract, principles that capture the ethos of FAIR vs specific principles that point to the means of implementation
- Sticking closely to the FAIR data principles or reinterpreting the FAIR ethos for software

Key assumptions:

- Application of the FAIR principles is the responsibility of the owner (often the creator) of the software, not the users
- Principles can be applied to any software used in research
- Software has a wide range of useful lifetimes, and findability, accessibility, interoperability and reusability will degrade over time. This shouldn’t be an excuse not to apply FAIR to software

Atelier - part 1

These slides https://tinyurl.com/RDA-France-FAIR4RS-slides

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

Q1: What is Research Software?

Summary report for subgroup3

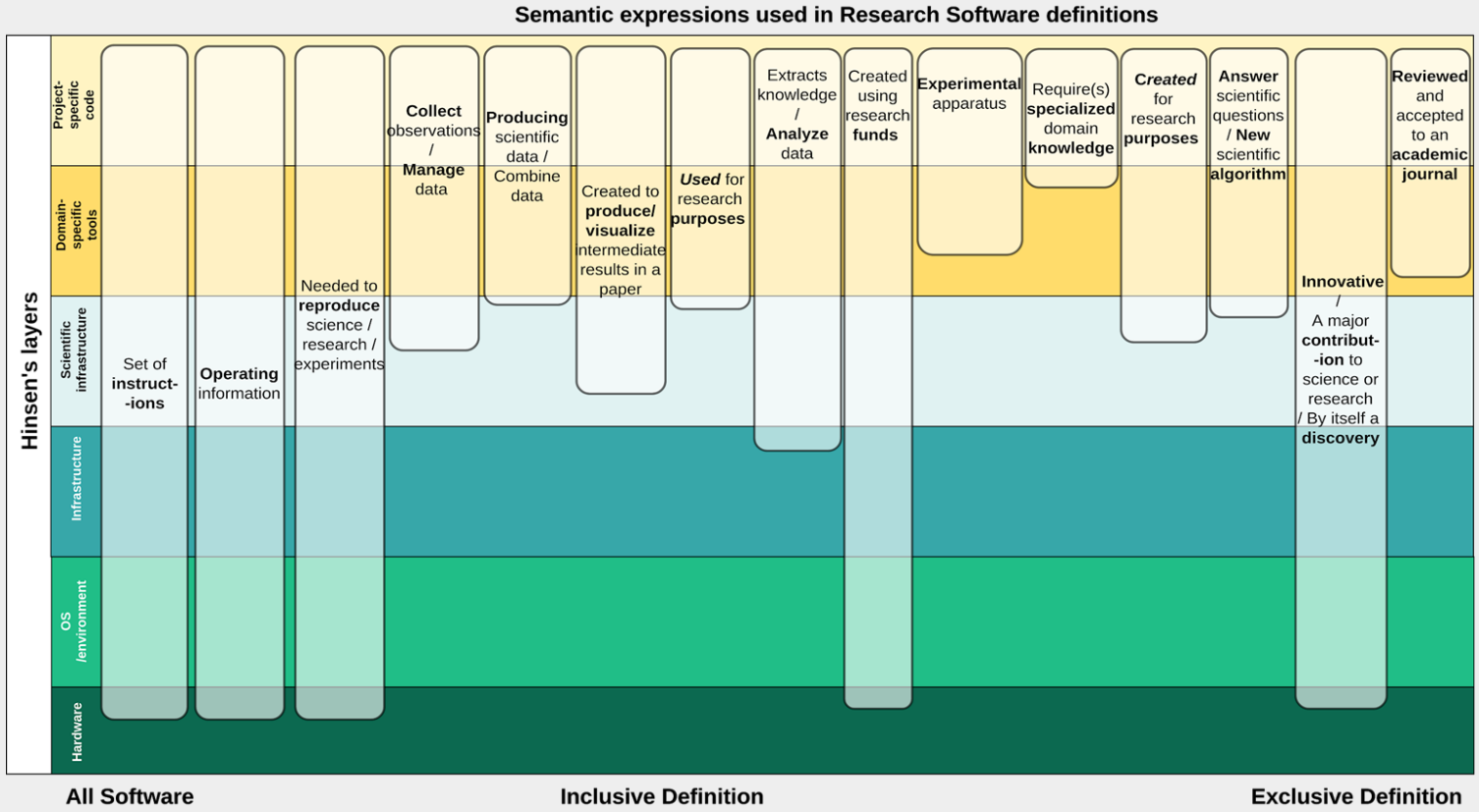
Controversy between:

- Inclusive definition
- Exclusive definition

Distinguishing

- Research software
- Software in Research

Collaborative notes
https://tinyurl.com/RDA-France-FAIR4RS



Defining the scope

Research Software definition (from the summary report of subgroup3 still in WG consultation)

> ***Research Software includes source code files, algorithms, scripts, computational workflows and executables that were created during the research process or for a research purpose. Software components (e.g., operating systems, libraries, dependencies, packages, scripts, etc.) that are used for research but were not created during or specifically for research should be considered software in research and not Research Software. This differentiation may vary between disciplines. The minimal requirement for achieving computational reproducibility is that all the computational components (Research Software, software used in research, and hardware) used during the research are identified, described, and made accessible to the extent that is possible.***

You can provide feedback on Q1.2 in the collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

FAIR principles for Research Software

These slides https://tinyurl.com/RDA-France-FAIR4RS-slides

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

FAIR Principles for Research Software -global

Findable: The software, and its associated metadata, should be easy to find for both humans and machines.

F1. Software is assigned a globally unique and persistent identifier

- F1.1. Different components of the software must be assigned distinct identifiers representing different levels of granularity
- F1.2. Different versions of the same software must be assigned distinct identifiers

F2. Software is described with rich metadata

F3. Metadata clearly and explicitly include the identifier of the software they describe

F4. Metadata are FAIR and are searchable and indexable

Accessible: The software, and its metadata, must be retrievable via standardized protocols.

A1. Software is retrievable by its identifier using a standardized communications protocol

- A1.1. The protocol is open, free, and universally implementable
- A1.2. The protocol allows for an authentication and authorization procedure, where necessary

A2. Metadata are accessible, even when the software is no longer available

Interoperable: The software interoperates with other software through exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs).

I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards

I2. Software includes qualified references to other objects

Reusable: The software is both usable (it can be executed) and reusable (it can be understood, modified, built upon, or incorporated into other software).

R1. Software is described with a plurality of accurate and relevant attributes

- R1.1. Software must have a clear and accessible license
- R1.2. Software is associated with detailed provenance

R2. Software includes qualified references to other software

R3. Software meets domain-relevant community standards

FAIR4RS WG. (2021, June). FAIR Principles for Research Software

FORCE11 The Future of Research Communications and e-Scholarship

<ReSA> Research Software Alliance

RDA RESEARCH DATA ALLIANCE

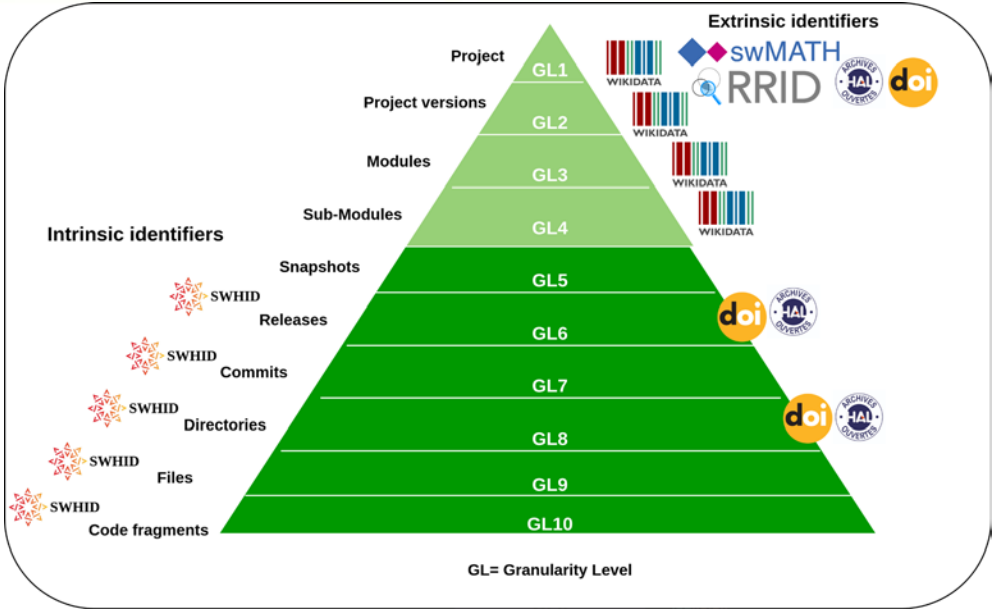
Findable

FAIR Guiding Principles (2016)	FAIR4RS Principles (2021)
F. The first step in (re)using data is to find them. Metadata and data should be easy to find for both humans and computers. Machine-readable metadata are essential for automatic discovery of datasets and services, so this is an essential component of the FAIRification process.	F. The software, and its associated metadata, should be easy to find for both humans and machines.
F1. (Meta)data are assigned a globally unique and persistent identifier	F1. Software is assigned a globally unique and persistent identifier.
	F1.1. Different components of the software must be assigned distinct identifiers representing different levels of granularity.
	F1.2. Different versions of the same software must be assigned distinct identifiers.
F2. Data are described with rich metadata (defined by R1 below)	F2. Software is described with rich metadata.
F3. Metadata clearly and explicitly include the identifier of the data they describe	F3. Metadata clearly and explicitly include the identifier of the software they describe.
F4. (Meta)data are registered or indexed in a searchable resource	F4. Metadata are FAIR and is searchable and indexable.

Q2: Software granularity and identifiers

Should the FAIR principles for research software care about the levels of granularity identifiers should be assigned?

If so, which are the most useful granularity levels to ensure the findability of software?



Extended figure from the SCID WG output http://doi.org/10.15497/RDA00053
Research Data Alliance/FORCE11 Software Source Code Identification WG, Software Source Code Identification Use cases and identifier schemes for persistent software source code identification (2020).

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

Accessible

FAIR Guiding Principles (2016)	FAIR4RS Principles (2021)
A. Accessible	
Once the user finds the required data, she/he needs to know how can they be accessed, possibly including authentication and authorisation.	**The software, and its metadata, must be retrievable via standardized protocols.**
A1. (Meta)data are retrievable by their identifier using a standardized communications protocol	A1. Software is retrievable by its identifier using a standardized communications protocol.
A1.1. The protocol is open, free, and universally implementable	A1.1. The protocol is open, free, and universally implementable.
A1.2. The protocol allows for an authentication and authorization procedure, where necessary	A1.2. The protocol allows for an authentication and authorization procedure, where necessary.
A2. Metadata are accessible, even when the data are no longer available	A2. Metadata are accessible, even when the software is no longer available.

Q3: Long term accessibility

Should software preservation be part of the FAIR principle?

“***A2. Metadata are accessible, even when the software is no longer available.***
Availability of software may change over time, because there is a cost to maintaining access or because the software has degraded and is no longer safely usable. The metadata describing the software is generally easier and cheaper to store and maintain than the software itself (e.g. in the software repository, or in a software registry or catalog) and there is value in understanding the details of the software even if it is no longer accessible. “

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Visit the archive

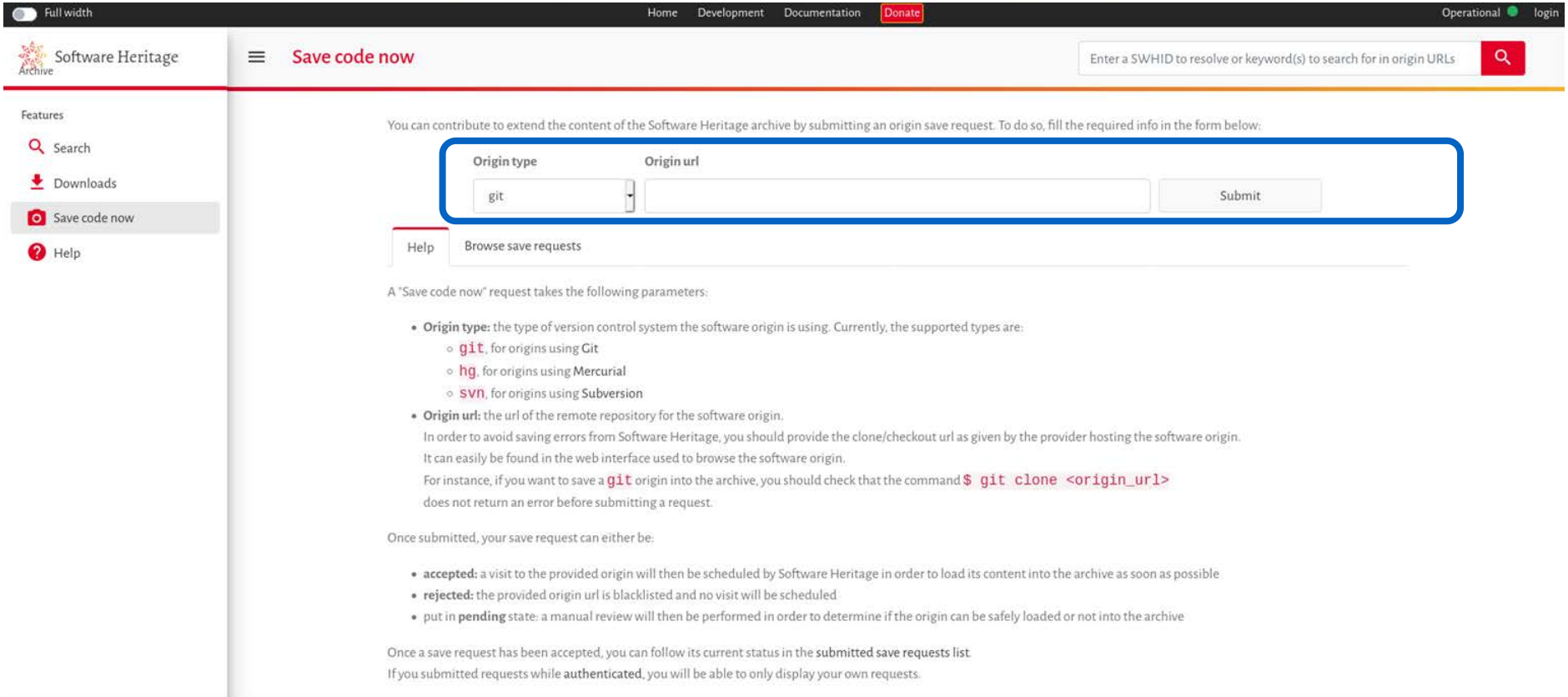
Collect, **preserve** and **share** all software source code

Preserving our heritage, enabling better software and better research for all

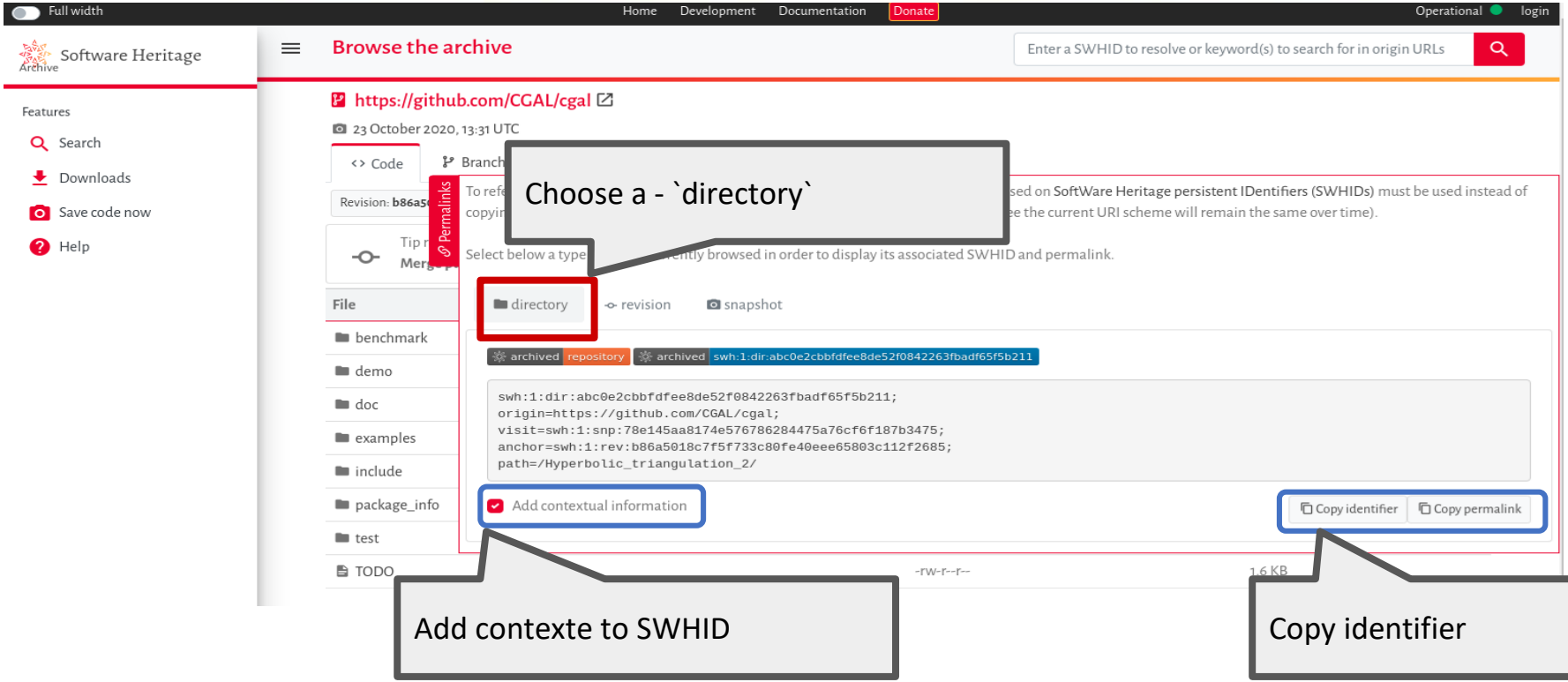
How it works?

- automatic pull from different forges (GitHub, GitLab, BitBucket),
- intrinsic metadata is extracted from the content itself,
- deposited artifacts are accepted only from known sources where metadata was moderated and curated
- Save Code Now feature to save **all public** git, svn and mercurial repositories
- SWHID persistent identifiers for **all the** source code artifacts

Save Code Now: you can save all public repositories



Choose a SWHID on Software Heritage



Break

FORCE11
The Future of Research Communications and e-Scholarship

<ReSA>
Research Software Alliance

RDA
RESEARCH DATA ALLIANCE

Atelier - part 2

These slides https://tinyurl.com/RDA-France-FAIR4RS-slides

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

Interoperable

FAIR Guiding Principles (2016)	FAIR4RS Principles (2021)
I. Interoperable	
The data usually needs to be integrated with other data. In addition, the data need to interoperate with applications or workflows for analysis, storage, and processing.	**The software interoperates with other software through exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs).**
I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.	I1. **Software reads, writes and exchanges data in a way that meets domain-relevant community standards.**
I2. (Meta)data use vocabularies that follow FAIR principles	
I3. (Meta)data include qualified references to other (meta)data	I2. Software includes qualified references to other objects.

Reusable

FAIR Guiding Principles (2016)	FAIR4RS Principles (2021)
Reusable	
The ultimate goal of FAIR is to optimize the reuse of data. To achieve this, metadata and data should be well-described so that they can be replicated and/or combined in different settings.	**The software is both usable (it can be executed) and reusable (it can be understood, modified, built upon, or incorporated into other software).**
R1. (Meta)data are richly described with a plurality of accurate and relevant attributes	R1. Software is described with a plurality of accurate and relevant attributes.
R1.1. (Meta)data are released with a clear and accessible data usage license	R1.1. Software **must have** a clear and accessible license.
R1.2. (Meta)data are associated with detailed provenance	R1.2. Software is associated with detailed provenance.
R1.3. (Meta)data meet domain-relevant community standards	R3. Software meets domain-relevant community standards.
	R2. Software includes qualified references to other software.

Another view: the EOSC SIRS TF (December 2020)

10.2777/28598

Software Infrastructure for Research Software Task Force: paves the way for federating the different approaches in view of supporting the software pillar of EOSC.

« the FAIR Guiding Principles for research **do not fit [software source code]** well, as they were not designed for it … » (FAIR does not fit publications either...)

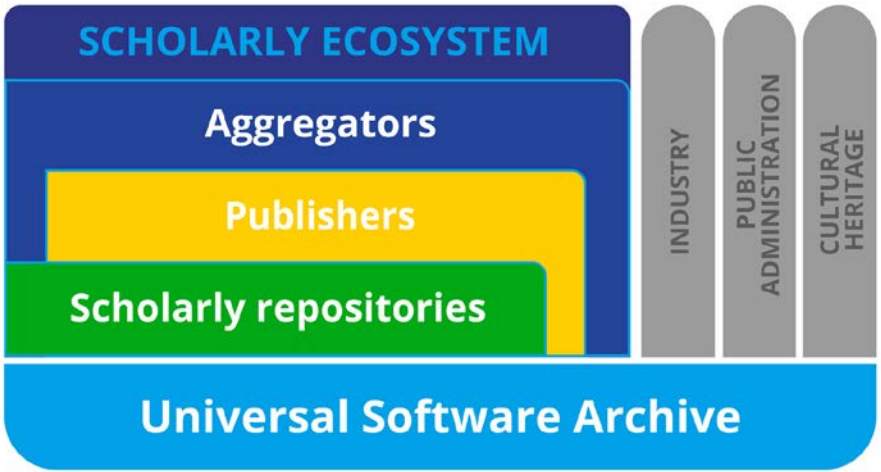
« We focus here on four key concrete issues that need to be tackled to make software a first-class citizen in the scholarly world, and where scholarly infrastructures play a prominent role: »

[**Archive**] ensure software artifacts are **not lost**

[**Reference**] ensure software artifacts can be **precisely identified**

[**Describe**] make it easy to **discover / find** software artifacts

[**Credit**] ensure **proper credit** is given to authors



Q4: Use and Re-use

Should FAIR software be re-executable?

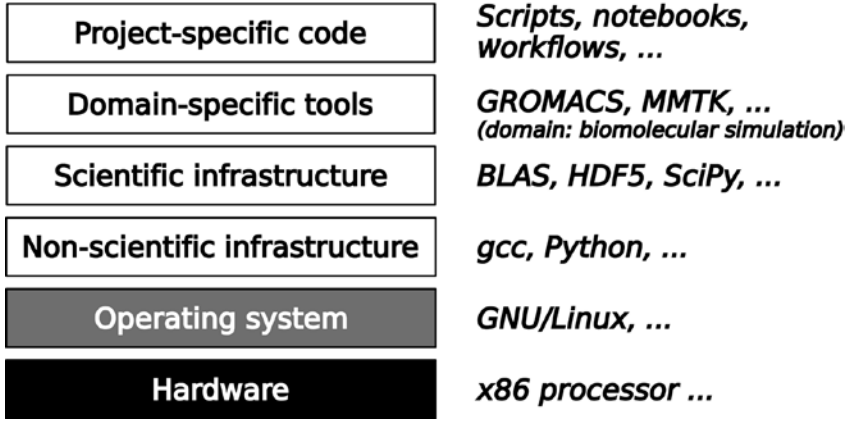
"Note that the general intent of these principles is that software is "executable in principle" - not "guaranteed to execute". Also, different aspects of reusability may best apply to different forms of software. For instance, source code might be modifiable but not executable without specialist infrastructure; libraries available as binaries can be built on and incorporated into other software but not easily modified. In general, source code is the most reusable form of software."

Q5: Software dependencies

Should FAIR be recursive?

Hinsen's software stack - from Dealing with software collapse:

1. "...software written by scientists for a specific research project...scripts, notebooks, and workflows, but also special-purpose libraries and utilities."
2. "...domain specific research software. These are tools and libraries that implement models and methods which are developed and used by communities ranging in size from a single research lab to thousands of researchers"
3. "...infrastructure created specifically for scientific computing, but not any particular domain."
4. "Infrastructure software that is not specific to scientific computing. … compilers and interpreters, libraries for data management, but also higher level tools such as text editors and Web browsers. … obtain[ed] from the wider non-scientific software market"
5. operating system
6. Hardware



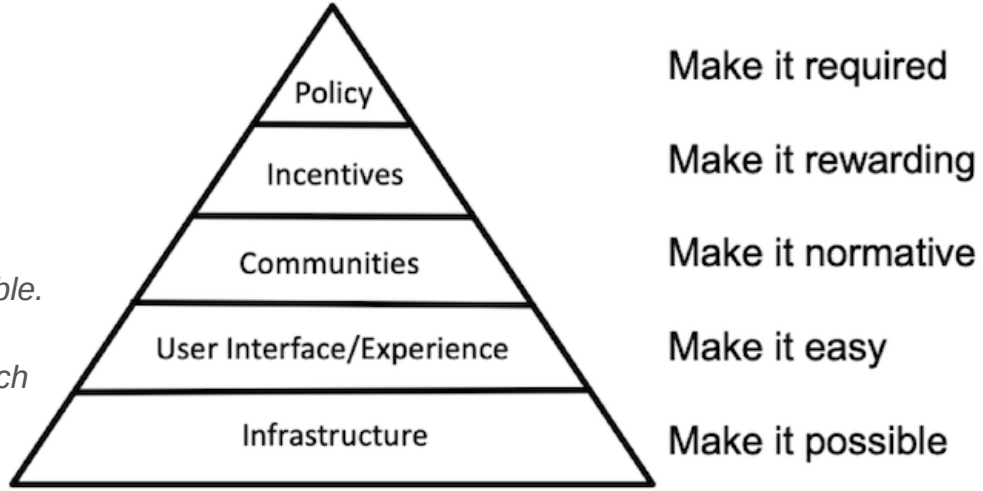
Konrad Hinsen. Dealing With Software Collapse. Computing in Science and Engineering, Institute of Electrical and Electronics Engineers, 2019, 21 (3), pp.104-108. hal-02117588

Q6: Who is responsible for FAIR software?

Who is expected to apply FAIR?

- And why?

*"The FAIR4RS Principles aim to provide guidance to **software creators and owners** on how to make their software Findable, Accessible, Interoperable and Reusable. The FAIR4RS Principles are also relevant to the larger ecosystem and various stakeholders that support research software (e.g., repositories and registries)."*



*"…the application of the FAIR4RS Principles is **the responsibility of the owners** (who are often the creators) of the software, not the users. "*

Challenges

These slides https://tinyurl.com/RDA-France-FAIR4RS-slides

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

Metadata

Metadata and identifier authority

“All research software must have unique identifiers and associated metadata. How are these identifiers created? How is the metadata created, stored and maintained? Intrinsic metadata, such as a codemeta.json file in the source code repository, is guaranteed to be controlled by the authors but must be exposed to make the software findable. Extrinsic metadata, such as (persistent) identifiers, can be used to make the software findable but is controlled by an external authority.”

Metadata vocabularies and metadata properties

“At present, there is no community agreement on which vocabularies should be used. Vocabularies used by package managers to describe software do not capture metadata about research and there are relatively few discipline-specific vocabularies that capture metadata about software development and usage. Establishing metadata vocabularies/standards is an intensive process for which resources are limited.”

Identification

Software identifiers

"At present, there is no community agreement on the best identifiers for software, even for specific use cases such as giving software authors credit. These identifiers are mostly independent and not clearly interoperable. This could be partly addressed through a community endorsement process, in one or more relevant communities."

Identification target

"At present, there is no community agreement on what a software identifier should refer to, e.g. for open source software, for commercial software, for a container, for a service, etc. This is discussed in the FAIR4RS Principles when talking about granularity and versions, and is also related to the idea of a software concept, which is the set of all specific versions of that software. Other work, such as (Hata et al., 2021), has highlighted challenges related to the linkage of scientific knowledge and software artifacts, e.g. in executable."

Complexity

Software structure complexity

"Software is often a complex object made up of other software, documentation, data and metadata, whose versions may change at different rates. How is this dealt with? Where should the FAIR4RS Principles be applied, and where should other interpretations of the FAIR Guiding Principles be applied? What should have identifiers, and how should relationships between them be described to be FAIR? Here, experience from applying the FAIR Guiding Principles to complex data collections may provide solutions."

FAIRness of related research objects

"There is still debate over whether FAIR is recursive, i.e. a digital research object is only ***"fully FAIR"*** *if the objects it builds on are also FAIR. However, even if just applied to data dependencies, this would restrict the implementation of FAIR4RS Principles as it would require measurable, actionable guiding principles to be applied down the complete dependency stack. This would ultimately be intractable as the authors of the software would not have responsibility for making the depende"*

Definitions

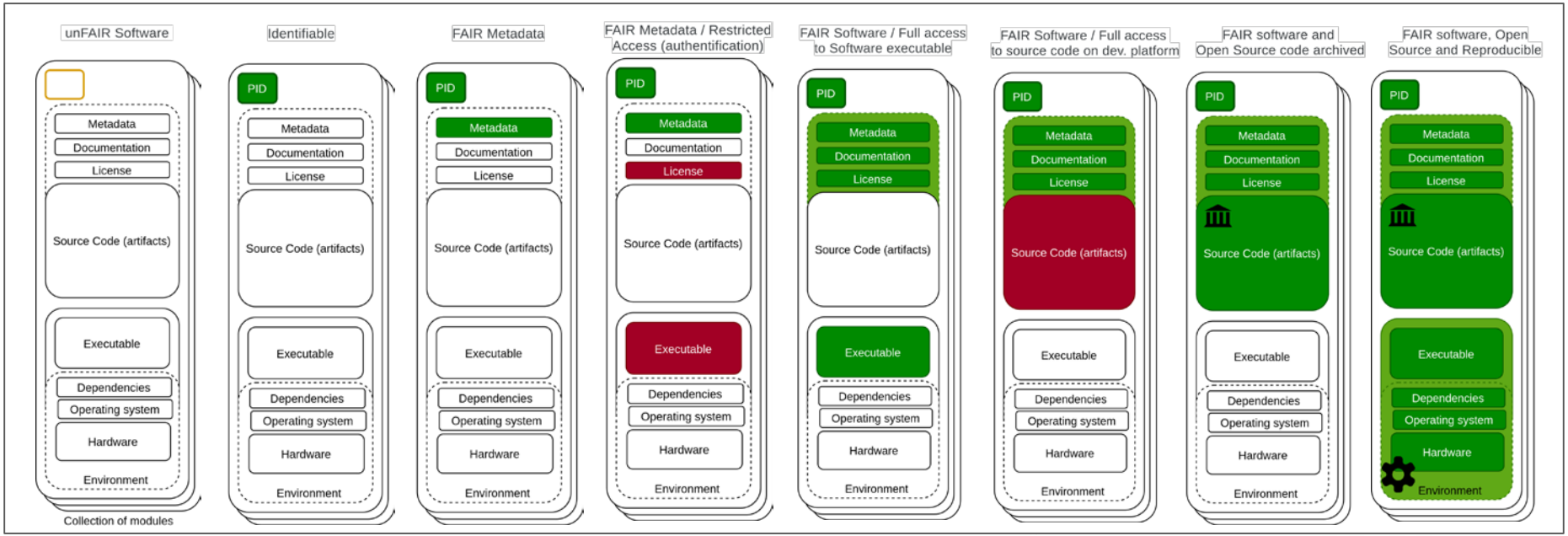
Definition of accessibility. In software engineering, there is already a different, well-understood definition of software accessibility. Even if the meaning used in the FAIR4RS Principles is well-defined and scoped, it may lead to confusion and mean the principle is not well-understood across all domains.

Definition of reusability. In software engineering, for software to be reusable it should also be maintainable and dependable (able to be built on for other purposes). This may be captured in R3, around domain-relevant community standards, but may also require additional clarification to avoid confusion or the proliferation of many competing sets of “added letters” to FAIR4RS related to other qualities.

Open source code

Openness and FAIR.

"Software does not need to be open source licensed to be FAIR: FAIR ≠ Open, Nevertheless, for software it may be easier to make it FAIR if it is open source. This can be seen in Figure 2 in Appendix A which summarizes software as ***increasingly FAIR research objects****. Open source software is generally more reusable, as the source code is accessible, and may be more interoperable because its APIs are inspectable. As with data, FAIR software should strive to be "as open as possible, as closed as necessary"."*



Q7: FAIR and FOSS

Should FAIR require software to have a free and open source (FOSS) license? Why or why not?

R1.1. Software must have a clear and accessible license.

"Software must have a license that clearly describes how it can be used and reused, ideally with conditions that are clear to humans and machines (e.g. using the specification published by the SPDX Consortium, 2020). To support a wide range of reuse scenarios, the license should be as open as possible. This license must also be compatible with the requirements of the licenses of the software's dependencies so that the software can be legally combined. "

Conclusion

These slides https://tinyurl.com/RDA-France-FAIR4RS-slides

Collaborative notes https://tinyurl.com/RDA-France-FAIR4RS

Consultation

- Open until July 11th
- On the RDA website
  - https://www.rd-alliance.org/group/fair-research-software-fair4rs-wg/outcomes/fair-principles-research-software-fair4rs

FORCE11
The Future of Research Communications and e-Scholarship
<ReSA>
Research Software Alliance
RDA
RESEARCH DATA ALLIANCE

Next Steps

- Publicize principles
- Encourage adoption, and highlight successful examples
- Develop curriculum and training
- Develop and encourage tooling to support applying principles
- Community work around gaps that prevent adoption
- Define metrics to measure adoption
  - For specific software
  - For principles and scholarly community as a whole
- Consider future governance of community and change processes

FORCE11
The Future of Research Communications and e-Scholarship

<ReSA>
Research Software Alliance

RDA
RESEARCH DATA ALLIANCE

Get involved!

- Join the RDA group and be part of the mailing list
- Come to events
- Follow the steering committee meeting minutes
- Say ‘Hi’ on the gitter channel
- Visit and read the publications on Zenodo
- Review the bibliography collected on Zotero

All this information is detailed on the community engagement channels page

Acknowledgements

- Working group chairs: Daniel S. Katz, Michelle Barker, Leyla Jael Garcia Castro, Neil Chue Hong, Jennifer Harrow, Carlos Martinez, Paula A. Martinez, Fotis Psomopoulos Morane Gruenpeter
- All ~215 members and contributors of the FAIR for Research Software Working group #FAIR4RS
- Present and past steering committee members for coordinating a range of activities
- Alfred P. Sloan Foundation and Wellcome Trust for support of the FAIR4RS WG

Thank you for joining

You can provide your feedback on https://tinyurl.com/RDA-France-FAIR4RS

Question?

morane@softwareheritage.org
@moraneottilia, @SWHeritage

https://www.softwareheritage.org/newsletter/

https://www.rd-alliance.org/groups/fair-research-software-fair4rs-wg