RDA VP17 FAIR for Research Software session (FAIR4RS)

Early outcomes of the FAIR 4 Research Software Working Group: definitions, principles and road ahead
See the time in your timezone | **Breakout 4**

Slides and notes

Presentation mode: https://tinyurl.com/FAIR4RS-VP17
Collaborative notes (this document):
https://tinyurl.com/FAIR4RS-notes-VP17
Index to these notes (you can also choose “View->Show document outline” to aid navigation:

- Participants list - add your name
- Icebreaker - place to add your answers
- Key questions on draft principles
- General notes on session
- Community engagement - add events FAIR4RS should present at
- Feedback - what could we improve?

Useful links

RDA group page, GitHub repository

How to 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

Meeting objective

At the beginning of July 2020, the FAIR for research software (FAIR4RS) WG started coordinating efforts to leverage community-led discussions on how to define and effectively apply FAIR principles to research software. This initiative has been divided across four subgroups with specific deliverables. We had about 50 participants who expressed interest and

contributed to one or more of these subgroups. During the Virtual Plenary 16 November 2020, the steering committee and the collaborators informed the broader community about the work progress in the last semester to gather feedback and initiate the next milestone work. This Plenary 17, we present an early draft of the principles based on the outcomes of all four subgroups, our plan for enabling broad community feedback, as well as outline the road ahead.

Participants

- ***Add your name, ORCID, member,*** *Click Join Group to become a member*
  https://www.rd-alliance.org/groups/fair-4-research-software-fair4rs-wg
  This is the preferred way for communication with the community.

Name *	***ORCID ****	***Are you a group member? (YES/NO)*** *You can join the group here https://www.rd-alliance.org/groups/fair-4-research-software-fair4rs-wg*
André Schaaff	0000-0002-8790-7048	Yes
Christian Pagé	0000-0001-7743-9570	Yes
Daniel S. Katz	0000-0001-5934-7525	Yes
Melissa Harrison	0000-0003-3523-4408	No
András Holl	0000-0002-6873-3425	Yes (I think)
Francoise Genova	0000-0002-6318-5028	Yes
Tom Honeyman	0000-0001-9448-4023	Yes
Sarah Davidson	0000-0002-2766-9201	yes
Leighton Christiansen	0000-0002-0543-4268	yes
Viviana Letizia	0000-0003-1088-5255	Yes
Wolmar Nyberg Åkerström	0000-0002-3890-6620	Yes
Fotis Psomopoulos	0000-0002-0222-4273	Yes
Anders Conrad	0000-0002-5283-2074	no
Victoria Dominguez Del Angel	0000-0002-5514-6651	Yes
Jen Harrow		Yes
Jez Cope	0000-0003-3629-1383	Yes
Susanna Sansone	0000-0001-5306-5690	yes

Anton Pirogov	0000-0002-5077-7497	no
Anna Mette Morthorst	0000-0001-5670-2464	no
Juan Bicarregui	0000-0001-5250-7653	Yes
Morane Gruenpeter		Yes
Paula Andrea Martinez		Yes
Hannah Mihai		no
Patricia Herterich	0000-0002-4542-9906	Yes (I think)
Hugh Shanahan	0000-0003-1374-6015	Yes
Marta Teperek	0000-0001-8520-5598	Yes
Zachary Trautt	0000-0001-5929-0354	No
Paul Stokes	0000-0002-7333-4998	No
Limor Peer	0000-0002-3234-1593	Yes
Kati Laakso	0000-0002-4160-3452	No
Julia Collins	0000-0003-4276-0318	Yes
Ants Finke		No
Siobhann McCafferty	0000-0002-2491-0995	Yes
Carlos Martinez-Ortiz	0000-0001-5565-7577	yes
Marie Timmermann		no
Leyla Jael Garcia Castro	0000-0003-3986-0510	Yes
Stéphanie Cheviron	0000-0001-8801-9133	Yes
Sara Studwell		No
Ikki Fujiwara	0000-0001-6841-5243	Yes
Mathieu Servillat	0000-0001-5443-4128	Yes
Thu-Mai Christian	0000-0002-3658-9692	Yes
Patrice Bellot	0000-0001-8698-5055	
Ray Plante	0000-0002-9279-4877	No

Andreas Pfeil	0000-0001-6575-1022	No
Gerard Coen	0000-0001-9915-9721	Yes
Victoria Moody		No
Ahmad Zainul Ihsan		No
Susanna Siri	0000-0003-3231-8139	No
Christophe Parisse	0000-0002-0010-3363	No
Tibor Kalman		Yes
Chris Hunter	0000-0002-1335-0881	no
Lauren Cadwallader	0000-0002-7571-3502	Yes
Hylke Koers	0000-0001-6538-7590	no
Susan Shingledecker	0000-0002-2088-8997	No
Violaine Louvet		No
Michelle Sibilla		No
Eefke Smit		
Mandy Gooch		
Harrison Dekker		
Janos Mohacsi		Yes
Franziska Zander		No
Martyn Rittman	0000-0001-9327-3734	
Jose Benito Gonzalez Lopez		
Becca Wilson	0000-0003-2294-593X	No
Céline Richard		No
Chris Erdmann	0000-0003-2554-180X	Yes
Josefine Nordling	0000-0002-6974-2825	Yes
Chirine Ghedira-Guegan	0000-0003-0908-2711	No

68 total

Ice-Breaker

Who is expected to apply FAIR? And why?

- PaulS: Researchers and Research Managers. They're the one who have the information for the metadata that enables FAIR and access to the systems that enable the metadata to be propagated
- Victoria: Research infrastructures
- Christian: Software developers but also anybody developing software for researchers (including researchers themselves). But more software developers than researchers will probably apply the FAIR principles. Why: to have more re-use of the efforts in developing those softwares, and more standardization of the coding process and outcome. This implies of course the "Findable" to be able to re-use software.
- András: Software developers (researchers) & software repository
- Patricia: mainly researchers to their data and software as quite a few of the principles are easier to apply early in the process. Service providers can help with infrastructure that make some of this really easy for them.
- Jez: Actually now I come to think about this, maybe it should be data stewards, curators, archivists with the long-term responsibility, but definitely facilitated by cooperation from creators/originators (authors, developers, researchers)
- Francoise: software developers in the context of projects, scientists developing s/w to be able to cite their development and for reproducibility. +1 for people who publish data, research, etc and data stewards
- Wolmar: All (publicly funded) constellations producing or managing data/software that support research and other outputs across society.
- Fotis: Software producer, research software engineers, researchers (managing software)
- Melissa: Researchers with support from those who are expecting them to apply - whether it be their institution, funder or publisher
- Limor: +1 Melissa
- VdA: software developers that need to find the right repositories, registres and PID's specific for software
- Paula: people who want their software to be reused and credited, acknowledged
- Sarah: People publishing in journals with requirements to publish relevant software
- Sarah: Software developers at institutions that require software data publication
- Sarah: Infrastructures publishing software (and participants in the infrastructure)
- Carlos: Software developers (RSEs) -- creators & current maintainers of software.
- Reusers

- Karsten: software developer, repository managers, data creators, authorities, trainers, FAIR evaluators themselves, lecturers,
- Morane: The infrastructures should propose assistance at the dissemination point when sharing/publishing software to make software FAIR, but only creators of the software can do these additions (e.g license)
- Dan: people who create software and the scholarly infrastructure that is used to publish/share/index it
- Repositories
- Viviana: Everyone, from researchers to policy makers
- LJ: Researchers, FAIR/RDM stewards, if apply = support then publishers, libraries, repositories, registries, archives
- Danie: data generators e.g. researchers, core facilities, informaticians
- Jen: Researchers and publishers, and data stewards
- Tom: Infrastructure providers (repositories, informatics capabilities/services), Software authors (researchers, research software engineers), policy makers (institutions, funders, publishers)
- Juan: Publicly funded researchers and research organisations
- Stéphanie : researchers, support staff, software developers,
- Marta: Researchers, Repositories, publishers, RSEs, anyone working with software :)
- Anders: researchers, software developers, data scientists/data stewards, FAIR software repository managers
- Siobhann: Researchers and RSE's
- André: R&D managers
- Ikki: Researcher who is responsible for the development of the software. Might be different from the developer/engineer who actually wrote the code under the contracts with the researcher. Because the findings brought by the software belong to the researcher.
- Author(s) of the resource, whether it's software or data, plus anyone curating the resources to make them available to a wider audience.
-
-

Key Questions on the Draft FAIR4RS Principles

Question 1: Domain-relevant community standards

In the draft FAIR for Research Software Principles we say that:

- To be interoperable, software reads, writes and exchanges data in a way that meets domain-relevant community standards
- To be reusable, software meets domain-relevant community standards

What does the term “domain-relevant community standard” mean to you? Can you describe the criteria you would use?

Name or identifier	**What does “domain-relevant community standard” mean?**	**What criteria do you use?**
Francoise Genova	In our case standards defined by the International Virtual Observatory Alliance (IVOA) (semantics, data access protocols, usage of the registry of resources to find data, etc) + data formats No standard for reusability of software themselves	Check compliance with the standards https://www.ivoa.net/documents/ Not sure we can readily check software in that respect :-) (no tool for that I think)
Christian Pagé	In the climate research community, we currently use the NetCDF file format (will probably evolve very soon, e.g. zarr or cubes), but also we have the CF-Conventions on the metadata and variable naming for example, that should be followed. Extending those conventions is essential also to support things like climate indices and indicators, which is being done in WGs. All those are very widely accepted in the research community and are necessary for intercomparisons.	CF-Conventions checker is an example (for compliance of the output of a given software). For publishing in the RI (e.g. ESGF), we have in the publisher a checker (CMOR) that must validate before publishing that the datasets follow the standards. Some tools and numerical models now output directly using the standards. That was not the case earlier.
Anton Pirogov	First thing I think of are file formats - these are essentially the “interfaces” of software. Also the kind of deployment usually used in that field (Github,	My subjective criterea when looking at a tool: Does the software support the major used file formats used for the

	special repositories,...?)	type of data? Is it easy to get the software and make it work? Are there unit tests in the project? Is it well documented?
Julia Collins	For interoperability (reading and writing data): Metadata schemas and data formats used for earth science data. (I.e. standards related to the data more than software-specific standards.) For reusability: Use of programming language(s) common in the community (e.g. Python, currently)	
Tom Honeyman	“Domains” could be research domains but also relevant professional domains (archiving, record keeping, software engineering), “domain-relevant” is potential domains of use. “Standards” in plural mean that there could be overlapping standards. “Community standard” implies that it can evolve, or reflect normative practice.	- Cited implementation guidelines - Certification processes - RDA WGs :-) - What do my peers do? (i.e., don’t go it alone)
András Holl	scientifi Used by the community and registered by the community as such (“standard”). Also documented.	
Sarah Davidson	Bio-logging and movement ecology: Not sure if there are any standards in my domain directly applying to software. Related questions might be: Does the software align with existing domain-relevant data standards/formats/platforms? Does it use common programming languages or software platforms in the domain? There might need to be some minimum sufficient or standardized documentation (readmes, comments, manuals) to enable understanding by practitioners in the domain.	In our use case (still in progress), we’ll be publishing only software that - aligns with (i.e. reads and manipulates) a domain-relevant data vocabulary/format/API - contains a required set of metadata - is provided with open licenses and in open formats Support will largely be for R-based software/scripts (widely used by researchers and practitioners in the domain)

Karsten Peters-von Gehlen	A standard which is specific to the research domain. Climate science is relatively advanced in this regard. E.g. netcdf file format, standardised lon-lat description, standardized time axis description, variable names are standardized (so that others know what others are talking about), standard ways of calculating specific diagnostics (often defined by specific papers)	In earth system science, standards apply to data. No domain specific standards for software exist per se. However, dedicated software packages or libraries exist for performing specific tasks or calculating specific diagnostics. This means, that using existing libraries/procedures ensures standard calculation. Standards applying to software apply to the capability of reading and writing data in the domain-relevant standards. Coding of complex numerical models on the other hand should follow computer science standards, though.
PaulS	Difficult question. Does domain refer to Comp Sci (programmer) domain or the User/researcher domain?	
Jez Cope		I would expect “domain” to be flexibly defined by the usage and intention of the originating researcher/developer(s)
Stéphanie Cheviron	In archaeology, standards like MIDAS or CARRARE, specific to a scientific community	
Carlos Martinez	Standards agreed by the research domain in question. For interoperable, this would mean using commonly used file formats. For reusability, this would mean being at a quality where other researchers in the same field would have the expertise to reuse the software with relatively low effort.	

Morane	Standards that are found in the community on how to archive, reference, describe, develop and cite software.	https://www.softwareheritage.org/save-and-reference-research-software Add the following files in the source code: - README - AUTHORS - LICENSE - (not required but recommended) codemeta.json
Wolmar	Widely-adopted and known by potential users of the software.	● Supported by data repositories ● ”Sufficient” specification and/or documentation to implement ● There is a community of maintainers for the standard
Juan	Rather than repeat the words from FAIR data and try to define them again, could we refer this back to FAIR data? So principle would be that S/W exchanges data that meets the relevant FAIR principles for Data.	
Anders Conrad	Widely used and supported wihtin e.g. discipline repositories, endorsed by standard bodies, cited in publications, etc.	Uptake, opposition, ability to translate to other formats, scientific bodies or reviews
André Schaaff	IVOA (...idem Françoise G.)	
Hugh Shanahan	What it isn’t is a single paper from a small set of authors with no evidence of wider use.	Evidence of uptake. Is there an associated web site with an active community forum?
chrish	a standard is a comminuty endorsed and upheld set of specific terms/protocols, but I think here you are trying to say community "norms" rather than specified standards. For the interoperability section the standards will likely be of the research domain for which the software has	

	been built to service. For the reuseable section the standards will be of the software domain to ensure the code is well formatted and constrained to community standards.	
Viviana Letizia	Why not ask the domain community as we ask for peer review standards?	
Esther Plomp	Any domain specific standards/vocabularies/formats that the community has set up or is frequently using and are relevant to software?	

Question 2: Should FAIR4RS include services and platforms?

Should the application of the principle of “accessible” be expanded to include instances of software running as services or platforms?

Name or identifier	**Should the application of the principle of “accessible” be expanded to include instances of software running as services or platforms? Give your reason.**
Ray Plante	Yes. From the standpoint of reproducibility, it can be helpful to others to return to the service and rerun the service. There are issues of the evolution of the service; nevertheless, it is useful to be able to requery or reanalyze with the service. As with anything with FAIR, it is an aspiration, and there will be limits on how accessible any component of an analysis workflow remains accessible.
Karsten Peters-von Gehlen	I am not sure how to answer this question. Is it about the code of the service or is it about the platform itself? The code should be FAIR, yes. However, I have problems getting my head around a service which is not “accessible”, as then it would not be a service.
Morane	No, a service (running software) is an instantiation of software which is dependent on the hosting platform of this service. The software behind this instantiation (if it is by itself Research Software) should be FAIR separately from its running form. If it is only software which is serving research, but not Research Software, it might act on data and enable FAIR. This is why services should be FAIR enabling and not FAIR as digital objects. For more information see the FAIRsFAIR output: FAIRsFAIR Report on Basic Framework on FAIRness of Services and was introduced to the research community at a dedicated webinar.
Fotis	No. The software that is need to run the service/infra needs to be FAIR, but the instance of the service does not necessarily need to have such a quality (as it may require additional data/digital object dependencies)
Victoria	No.If we have already the source code and metadata
Anders	Not as a requirement. This would hugely limit the scope and usefulness. And limit to financially resourceful parties, commercial companies.
Julia	Yes. Accessibility to me applies to availability of the source code, not the software in a “running” state. Therefore a service is no different than any other piece of software -- what needs to be accessible is the source, not necessarily the compiled and running application. Or, does

	“services and platforms” refer to the environment in which the software is being executed? If so, that feels like ancillary information that could be described (perhaps the description is a requirement…), not something that needs to be included in the accessibility requirement.
Andras Holl	Yes. One may think that services are necessarily accessible, per definitionem. But what about long term accessibility?
Carlos Martinez	Yes, software which relies on external services to run will not be able to function without these services.
Anton Pirogov	Yes, today much software is published in form of microservices etc. and interacts using internet protocols. Also, some software is too difficult or infeasible to set up locally, so most users will probably use a hosted instance somewhere in this case. Unfortunately some services appear and vanish shortly after, that must change.
PaulS +1	Depends on whether the software is inextricably linked to the platform or not.
Juan	No, not in the same definition. If we broaden the scope it will be more difficult to be precise in the principles. But we should cover it elsewhere - Instead could add a reference to another definition for services and platforms (ie work on that separately)
Hugh	I would say No as the level of difficulty expands enormously - what happens with updating a service? Do we have to start worrying about service availability? Also is the service data or software (often this is mixed up) and hence you know have to ask which of the FAIR principles apply (okay head hurting now)?
Ikki	Yes. Because services are more fragile (subject to change) and short-living, while possibly being an essential part of the research tool chain. Without FAIRness of services the research will not be reproducible. I think, for a service to be FAIR, its source code and runtime environment should be reproducible by others.
Wolmar	No, not as a requirement. The most effective way to provide a service will change over time as financing conditions and usage patterns change. As an alternative, maybe a reference deployment should be described.
Jez	This is a really tricky question! My gut says yes but I’m not sure how in practice. I would certainly want to try and make workflows (that combine and orchestrate multiples services for a speecific outcome) should be FAIRable, but then is that meaningful if the services referred are *not* FAIR?

Francoise	From the end user point of view I would say that one needs to be able to find and access a service. FAIR for services is an issue by itself, described as FAIR-enabling software, not sure it should be taken into account here - risk of lack of focus because the issues can be different (additional elements in the definition of a service and somehow different expectations).
Chrisian Pagé +3	No, I think it should be separate to not widen the scope too much and have to accommodate too many different things in FAIR4RS.
Sarah Davidson	Might depend on whether the service/platform is required for the use of the software. Or whether the software is designed/documented in a way that allows it to be potentially deployed on or modified for other services/platforms.
Michelle Sibilla	We have to separate software development and deployment. We can capitalize and enhance code in FAIR libraries
Mathieu Servillat	No, services may use FAIR software and so should be more trustable, but FAIR for a service sounds too different than for software
Janos Mohacsi	Software yes, but the instance might be decided by the instance owner.
Viviana Letizia	Yes the public one, and recommended the private ones of particular interest
Tom Honeyman	Genuinely unsure. I can't think of a principled reason that it should be in or out, just that whichever choice is made, it will have significant ramifications. Data services seem to be in scope for the FAIR principles for Data, so if software services were in scope for FAIR4RS, differentiating data and software services would be one of the things that would need to be resolved. Second, this interacts with the granularity and versioning questions. A software service is a manifestation of a software project. Cf the discussion in the data granularity group… this kind of "versioning" has not been considered for software yet, which has focussed on source code, not expression or manifestation dimensions. The informatics needs are quite different between source code and services. Services are ephemeral and involve the allocation of resources to maintain (although there is an analogous consideration in software infrastructure, e.g., the human infrastructure needed to keep the software maintained).

	Finally if we exclude services, then I think it is worth considering whether it's actually "FAIR for research software source code", not "research software" because the label alone is super clear.
Stéphanie Cheviron +5	Tricky question. If the service/platform is essential to run the source code, yes. If not, no.
André Schaaff	in a perfect world the softwares should imply it for their instances, the contrary is not true, so globally no

Question 3: References to other objects

What types of objects must software be able to interact with, and therefore need to be referenced? Can you provide examples of the different types of objects? Should the references to these objects be part of interoperability, reusability or both?

Name or identifier	**What types of objects must software interact with?**	**Interoperability, reusability or both (with your reason)?**
PaulS +1 +1 +1 +1	Operating system +1 Hardware +1 (processor and peripherals) Linked libraries +1 Programming framework (infers programming language) Defined Versions of all of the above Inputs	Mainly reusability, with overtones of interoperability
Fotis +1	All digital objects need to be references	Digital objects should be under interoperability. Software (as a subset of Digital Objects) fits under reusability
LJ	Data, other software, workflows	

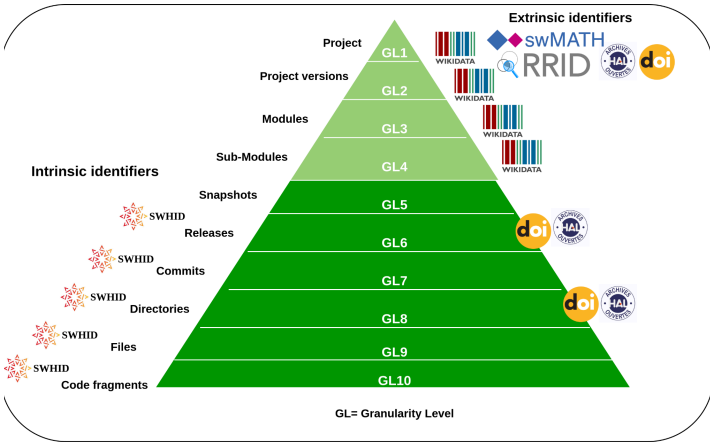
	2	
András Holl	Data. Which should have DOI.+1	
Morane	All objects (research or not) on which the software at hand is related to or dependent on. It can be articles, data or other software. +1 +1	Both but with some distinction between which reference is reviewed
Anders	Running code or metadata to software? I don't understand...	
Paula	Software dependencies, different versions of the same software (to build upon previous versions)	More into interoperability, so that different software can communicate. But also Reusability to the person who is reusing the software is aware of the components that are required.
Janos Mohacsi	Tested Operating system, HW requirements, dependencies, software frameworks	reusability
Karsten	IT environment, data, PIDs handled in the system if a workflow framework is used, other software, libraries,	Reusability is key. Achieving interoperability in the sense that it runs on different systems, is too hard. Running it in a containerized framework is more straight-forward. Definition of interoperability is not clear here.
Wolmar	Host system, drivers and hardware API:s. +1+1 External web service API:s.	Interoperability ● Plugin system / configuration utilities to support running the software in new environments – potentially mapping to other host / external API:s. Reusability ● External requirements

		to set up the software needs to be well documented and defined. ● Add system checks to ensure that the host environment and web services are available and compatible?
Melissa	Research publication (article) and data and other software required to run it/process etc (dependencies)	
Carlos	Other software (dependencies, including libraries & operating system) and data necessary for running the software (whether this is data “bundled” together with the software or available from an external source (service).	Both.
AndreasP	PIDs in general. Ideally extending the “open file” dialogue with a “open PID” dialogue.	Both?
Julia	Operating system, language version, versions of libraries or packages	
Limor +1	Any objects needed to reproduce a scientific claim	both
Ikki	In addition to the dependencies (libraries, packages, etc.), the runtime environment (including middleware and hardware) should also be referred.	Both
Juan	Yes, right to split the two cases. All of the above *may* need to be referenced. None are compulsory. Also relevant is the rich metadata principle from Findable. Some of the things above are about finding whether the S/w is any use to you. So are about findability.	

Question 4: Software granularity and identifiers

We currently say: “F1. Software is assigned a globally unique and persistent identifier.”

The figure on the right is an extended version from the SCID WG output[1].



Should the FAIR principles for research software care about the levels of software granularity?

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

Name or identifier	**Should the FAIR4RS care about the levels of software granularity? If so, which are the most useful granularity levels to ensure the findability of software?**
Francoise Genova +1	Granularity is a real issue for the FAIR principles when applied to data also, since it fully depends on the use case. Will the software case have an easier answer??? Retrieve an executable software? Going down to modules for reproducibility? But increases the ‘interoperability’ issues when one goes to finer grain.
PaulS	YES! As to the level… The level at which results can be reproduced/broken (which is probably releases). However, what about forked open source. Where the fork works for this use case and the core doesn’t.
Janos Mohacsi	Yes. level depends. The level that allows identifying a particular environment. If the environment patched individually, then code fragments. If use releases, project versions, than that level….
Michelle Sibilla	does granularity deal with dependency ?
András Holl	As for reproducibility you should go for the most detailed level, and

[1] 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).

	Maybe not only the exact code, but the parameters used and maybe even the environment.
Anders Conrad	The “useful” level of granularity depends on the use case. I don’t think this should be specified in principles, as little as it is in original FAIR principles, policies for PIDs, etc. It is an implementation issue.
Anton Pirogov	Yes, and from the user point of view, I’d say realistically is should be down to releases (up to GL6). More granular versioning should also exist, but it is mainly for developers. The user ultimately installs some tagged release in most cases, and this is what matters for “findability” from the outside to reproduce something etc.
Hugh Shanahan	Yes - granularity is useful. OTOH I think we only need to get to the point of releases GL1-GL6
Victoria	YES !!! GL1-GL7
Morane	Metadata and software description are at the higher levels of the granularity pyramid and should be associated with an extrinsic identifier. => findability The lower levels are needed for reproducibility. => reusability So yes, Granularity levels should be acknowledged even if it is not required by the FAIR principles in particular.
Fotis	Granularity does make sense to be connected to FAIR, but not as a blanket connection. Especially when looking a finer level of granularity, and therefore not really useful beyond a rather narrow context (such as the actual developer/s)
Juan	All levels - up to the author to decide what might be useful to others - but that makes for a lot of work in providing metadata so maybe unrealistic.
Carlos	Yes -- all different levels have different objectives, so I would say it is dependent on the use case. For reuse, a low level of granularity (maybe commit level?) would be necessary; for credit a higher level of granularity (project?) may be good enough.
Wolmar	Yes, in the sense that the principles should make it clear that there exists many levels of granularity and that it should be an active choice that makes sense to the particular software and the potential audiences using/referencing it.
Paula	Yes, Project version seems a viable granularity to make FAIR
AndreasP	All mentioned levels of PIDs should exist as needed, but I do not see how or why they should be defined in the FAIR principles.

Chris	There is another aspect to consider when talking to James Howison and others was that there was a training/collaborative use case. James mentioned being able to reference at the pull request level. Might be a stretch but I believe I've seen this possibly in The Carpentries.
LJ	Not necessarily (at principle level) but I would be in favour to include project (but this one is a changing object so not sure FAIR fully applies) and released version
Stéphanie Cheviron	Yes, at level 6 and 1. I don't think that below level 6 is necessary.
Mathieu Servillat	As for FAIR data, granularity will depend on use cases, and for a given level, the application of FAIR principles may imply different solutions : provenance of a commit is different from the provenance of a package or a project… identifiers for levels < 6 might be related to level 6 identifier. At least, GL for RS seem to be more easily defined!
Ikki	Wherever level at which the reproducibility of the research is guaranteed. Maybe at Level 6.
Tom Honeyman	I think this is an implementers concern. Granular citation is a service feature for consumption not a requirement for depositing. Moving a piece of software into infrastructure that supports it makes that feature available for that software, but not all software requires this. It also places a barrier on adoption. This makes it harder to apply the FAIR principles. Finally, if services are considered in scope then the SCID WG output is insufficient as it focuses on source code. How do we talk about two different deployments (manifestations)? Also how do we talk about algorithmic implementations across different languages (expressions)?
Esther Plomp	Yes, granularity is important: releases seem to be a good level, below that is probably too detailed and too difficult to keep track of in terms of FAIR.

Question 5: Relationship of FAIR4RS to FAIR data principles

Should the FAIR principles for research software be standalone so that they can be read and implemented independently of knowledge of the FAIR data principles (noting that this may repeat parts of the FAIR data principles, particularly in the FAIRness of metadata)

OR

Should principles that are directly taken from the FAIR data principles (e.g., around the FAIRness of metadata) be removed from the FAIR software principles in favor of a statement that these principles complement but do not replace the existing FAIR data principles?

Name or identifier	**Standalone or remove replicated principles - give your reason**
Francoise Genova	Have the FAIR data principles for software usable standalone.
Anders Conrad	Standalone and complete, for possible use by software community
PaulS	Standalone. Referring back will muddy comprehension (witness the discussion here with nuanced understanding)
Patricia	Standalone (maybe with footnote where the overlap to data is)
Janos Mohacsi	Standalone - different group will use.
Morane	Standalone
Chris	Standalone
Stéphanie Cheviron	Without hesitation: standalone.
Anton Pirogov	Standalone so it is self-contained, but maybe with highlighting of the additional/changed points, for people already familiar with the general principles, so it is easy to see “what’s new”
LJ	Standalone (it would make it easier for maturity models, metrics, etc. to evolve on their own) but still can be connected (with references to each other)
Victoria	Standalone !!
Leighton +1	The FAIR4RS should be stand-alone with reference or link to data principles.
Wolmar	Standalone. In a way that it can be put in the hands of a research Software developer with little to no initiation to the FAIR data

	community.
Julia	Standalone, but will need a plan to ensure duplicated parts are kept up to date if the FAIR data principles evolve.
Ikki	Standalone. Self-descriptiveness is always helpful.
Mathieu	Standalone, the objective of a software may not be to deal with FAIR data
Esther	Standalone, otherwise it will be confusing and complicated to look up things, especially if the data/software principles will diverge in the future.

Notes

How to 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

Future work - community engagement plans (3 minutes)

- What other events would benefit from FAIR4RS workshops?
- Can we partner with your organization to enable consultation in your community?

please add your suggestions below with contact details if possible here are the current and past activities from the WG https://github.com/force11/FAIR4RS/blob/master/CommunityEngagement.md	+upvoters
Is anyone going to/proposing a talk at Announcing the Global Maintainer Summit - The GitHub Blog (Chris)	

Feedback

Thanks for joining us!!!

Let us know your thoughts of this session, we are keen to improve.

- Excellent session!
- Yes, this was a really great session! Informative + interactive
- Got a great way to have a remote interactive session!
- Really well run session. Good structure and pace.
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

FAIR Principles (as background)

FINDABLE:

F1. (meta)data are assigned a globally unique and eternally persistent identifier.
F2. data are described with rich metadata.
F3. (meta)data are registered or indexed in a searchable resource.
F4. metadata specify the data identifier.

ACCESSIBLE:

A1 (meta)data are retrievable by their 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 data are no longer available.

INTEROPERABLE:

I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. (meta)data use vocabularies that follow FAIR principles.
I3. (meta)data include qualified references to other (meta)data.

REUSABLE:

R1. (meta)data have a plurality of accurate and relevant attributes.
- R1.1. (meta)data are released with a clear and accessible data usage license.
- R1.2. (meta)data are associated with their provenance.
- R1.3. (meta)data meet domain-relevant community standards.

Chat log