

Toward Better Research Software

Daniel S. Katz

Chief Scientist, NCSA

Associate Research Professor, CS, ECE, iSchool

d.katz@ieee.org

[@danielskatz](#)

I ILLINOIS

NCSA | National Center for
Supercomputing Applications

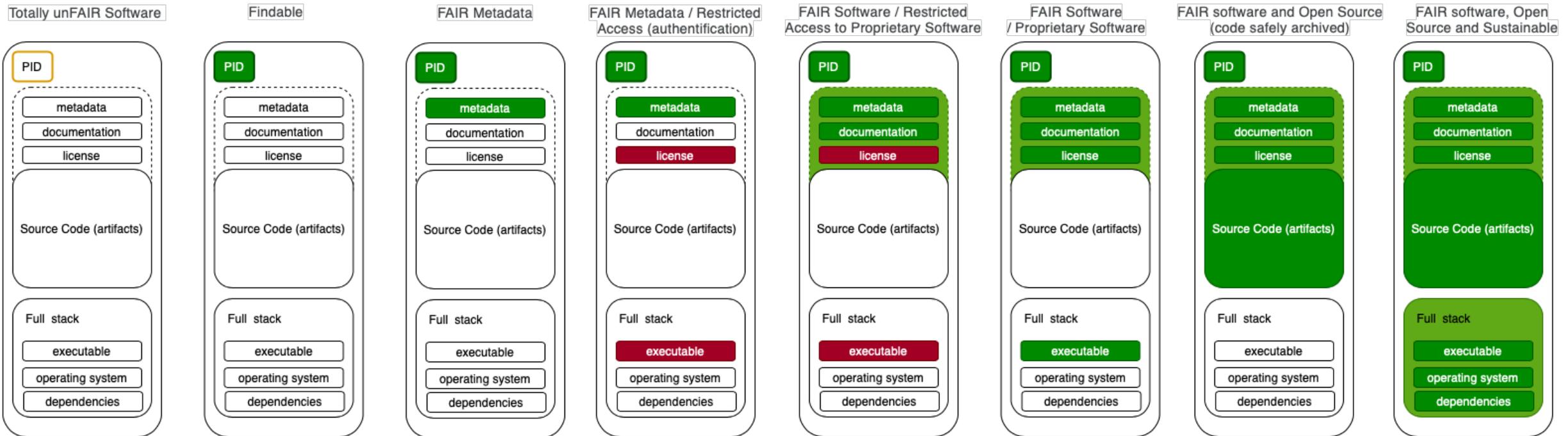
DOI: [10.5281/zenodo.4551441](https://doi.org/10.5281/zenodo.4551441)

Vision

- All research is reproducible
- All research software is open
- All research software is usable by others (for their own research)
 - And is cited when it is used
- All contributors to research software are recognized for their work
 - With good careers
- All research software is sustained as long as it is useful
- All research software is high-quality and robust

Implementing parts of the vision - FAIR

1. Making research software Findable, Accessible, Interoperable, and Reusable (FAIR)



FAIR for Research Software (FAIR4RS)

- Working group defining FAIR principles for research software
- Led by Morane Gruenpeter, Paula A. Martinez, Carlos Martinez, Michelle Barker, Daniel S. Katz, Leyla Garcia, Neil Chue Hong, Fotis Psomopoulos and Jennifer Harrow
- Timeline
 - July 2020 – Feb 2021: subgroups
 - March 2021 - Complete first draft of principles
 - April - June 2021 - Engage community around this draft
 - July 2021 - Finalize principles



FAIR for Research Software (FAIR4RS) subgroups

1. 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 (e.g., Lamprecht et al. 2020. Towards FAIR principles for research software. <https://doi.org/10.3233/DS-190026>)
 - Published: Katz DS, Gruenpeter M, Honeyman T, et al. (2021). A Fresh Look at FAIR for Research Software. arXiv:2101.10883 [cs.SE], <https://arxiv.org/abs/2101.10883>
2. FAIR work in other contexts
 - Analyzed how FAIR principles are applied to research objects other than data/software – final report
3. Research software definition
 - Reviewing existing definitions and to specify the scope for the WG outputs – in progress
4. New research related to FAIR Software
 - Review recent research and studies around FAIR software
 - Via up-to-date identification of approaches that can help structure FAIR4RS work, in form of Zotero reading list and short report on important insights from review and survey – in progress

Next steps: complete all subgroups, merge into combined draft report

Implementing parts of the vision - recommendations

- 4 Simple recommendations for Open Source Software - a Software Carpentry module
 1. Make source code publicly accessible from day one
 2. Adopt a license and comply with the license of third-party dependencies
 3. Define clear and transparent contribution, governance and communication processes
 4. Make software easy to discover by providing software metadata via a popular community registry
- Five recommendations for FAIR software
 1. Use a publicly accessible repository with version control
 2. Add a license
 3. Register your code in a community registry
 4. Enable citation of the software
 5. Use a software quality checklist
- Software Sustainability Institute publishes various guides, including for developers, and an on-line software sustainability evaluation
- Better Scientific Software (BSSw) also collects and shares resources about software, including for planning, developing, performance, reliability, collaboration, and skills

Implementing parts of the vision - info from reviews

- JOSS review criteria

- License
- Documentation
- Installation instructions
- Example usage
- API documentation
- Community guidelines (contributing, reporting, support)
- Verified installation, functionality, performance
- Tests
- (implied) appropriate packaging

- CZI EOSS award statistics

Essential Open Source Software RFA by the Numbers

Open Source Projects:

475
applied

42
funded

Project with	Applied	Funded
documentation	90%	100%
an issue tracker	88%	95%
continuous integration	60%	90%
a community forum	57%	91%
contributor guidelines	52%	84%
package distribution	47%	80%
a Code of Conduct	46%	70%

Conclusions

- We know a lot about how to make research software, and the research that uses it, better
- We now need to choose to do the work
- For people who really want their software to be widely used, making it FAIR and open is a good path
- Recommendations exist to help ease the path for others to contribute, leading to community software