PRACTICAL SOFTWARE CITATION

FOR RESEARCH SOFTWARE DEVELOPERS, MAINTAINERS AND USERS

Stephan Druskat
German Aerospace Center (DLR), Institute of Software Technology

DOI 10.5281/zenodo.15583669

DLR

About me

- M.A. *English Philology*, *Modern German Literature*, *General and German Linguistics* (Free University Berlin)
- ~13 years *RSE* in several linguistic research projects
- Fellow of the Software Sustainability Institute
- Lead Citation File Format project
- Since 2019: SE researcher at DLR (*Sustainable Software Engineering*)
- PI HERMES project (*automating software publication with rich metadata*)
- Chair ReSA Task Force "Software Authorship & Contribution"

Overview

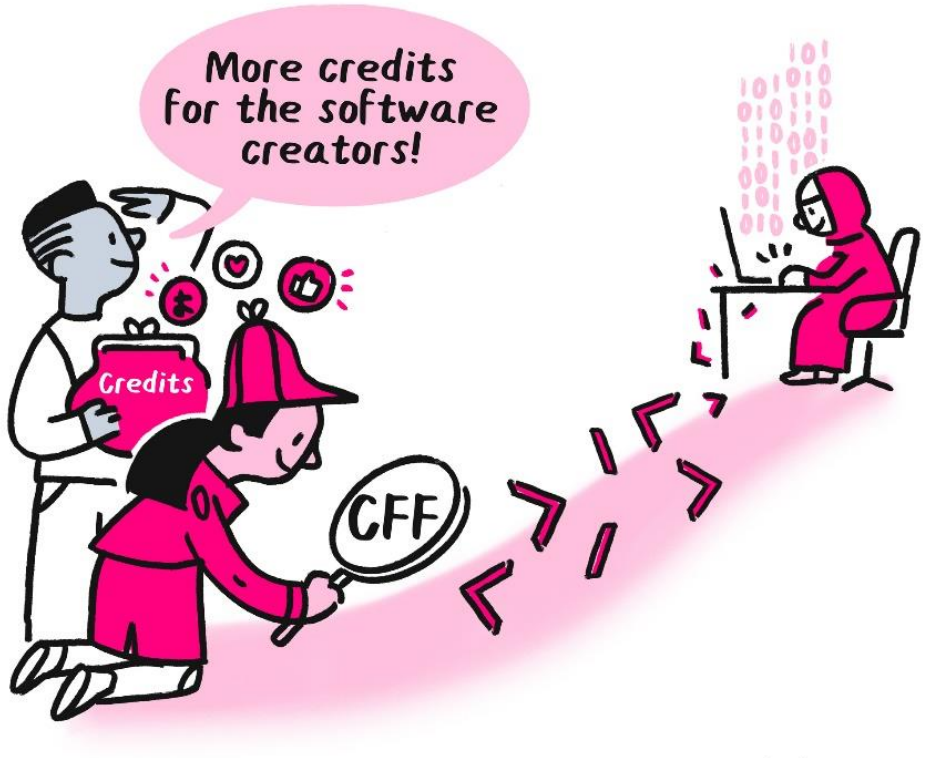
- Software citation
- Practical software citation for developers and maintainers
- Practical software citation for software users
- Wrap-up

SOFTWARE CITATION

Bildquelle hier angeben

Why software citation?

- Principles of research apply: disclosure of research process
- Accountability: taking credit
- **Formal citation** as established method:
  - **Importance** of software in research
  - Findability (**FAIR**)
  - **Reproducibility**
  - **Credit**
  - …



More credits for the software creators. The Turing Way project illustration by Scriberia. Zenodo. https://doi.org/10.5281/zenodo.3332807
License: CC BY-4.0

The principles of software citation

Software citation principles

Arfon M. Smith[1,*], Daniel S. Katz[2,*], Kyle E. Niemeyer[3,*] and FORCE11 Software Citation Working Group

[1] GitHub, Inc., San Francisco, California, United States
[2] National Center for Supercomputing Applications & Electrical and Computer Engineering Department & School of Information Sciences, University of Illinois at Urbana-Champaign, Urbana, Illinois, United States
[3] School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, Oregon, United States
* These authors contributed equally to this work.

ABSTRACT

Software is a critical part of modern research and yet there is little support across the scholarly ecosystem for its acknowledgement and citation. Inspired by the activities of the FORCE11 working group focused on data citation, this document summarizes the recommendations of the FORCE11 Software Citation Working Group and its activities between June 2015 and April 2016. Based on a review of existing community practices, the goal of the working group was to produce a consolidated set of citation principles that may encourage broad adoption of a consistent policy for software citation across disciplines and venues. Our work is presented here as a set of software citation principles, a discussion of the motivations for developing the principles, reviews of existing community practice, and a discussion of the requirements these principles would place upon different stakeholders. Working examples and possible technical solutions for how these principles can be implemented will be discussed in a separate paper.

Subjects Digital Libraries, Software Engineering
Keywords Software citation, Software credit, Attribution

SOFTWARE CITATION PRINCIPLES

The main contribution of this document are the software citation principles, written fairly concisely in this section and discussed further later in the document (see Discussion). In addition, we also motivate the creation of these principles (see Motivation), describe the process by which they were created (see Process of Creating Principles), summarize use cases related to software citation (see Use Cases), and review related work (see Related Work). We also lay out the work needed to lead to these software citation principles being applied (see Future Work).

1. **Importance:** Software should be considered a legitimate and citable product of research. Software citations should be accorded the same importance in the scholarly record as citations of other research products, such as publications and data; they should be included in the metadata of the citing work, for example in the reference list of a journal article, and should not be omitted or separated. Software should be cited on the same basis as any other research product such as a paper or a book, that is, authors should cite the appropriate set of software products just as they cite the appropriate set of papers.

Submitted 24 June 2016
Accepted 23 August 2016
Published 19 September 2016

Corresponding author
Daniel S. Katz, d.katz@ieee.org

Academic editor
Silvio Peroni

DOI 10.7717/peerj-cs.86

How to cite this article Smith et al. (2016), Software citation principles. Peerj Comput. Sci. 2:e86; DOI 10.7717/peerj-cs.86

A. M. Smith, D. S. Katz, K. E. Niemeyer, and FORCE11 Software Citation Working Group, "**Software citation principles**," *PeerJ Comput. Sci.*, vol. 2, no. e86, 2016, doi: 10.7717/peerj-cs.86.

1. **Importance**

   Software is cited like papers are cited.

2. **Credit and attribution**
3. **Unique identification**
4. **Persistence**
5. **Accessibility**

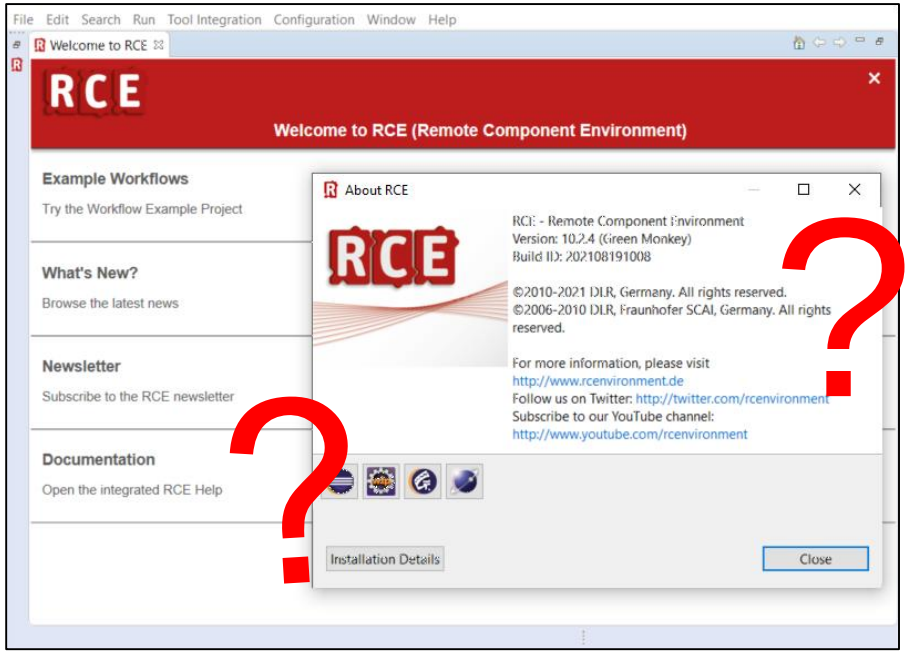
   Citation allows access to software and metadata.

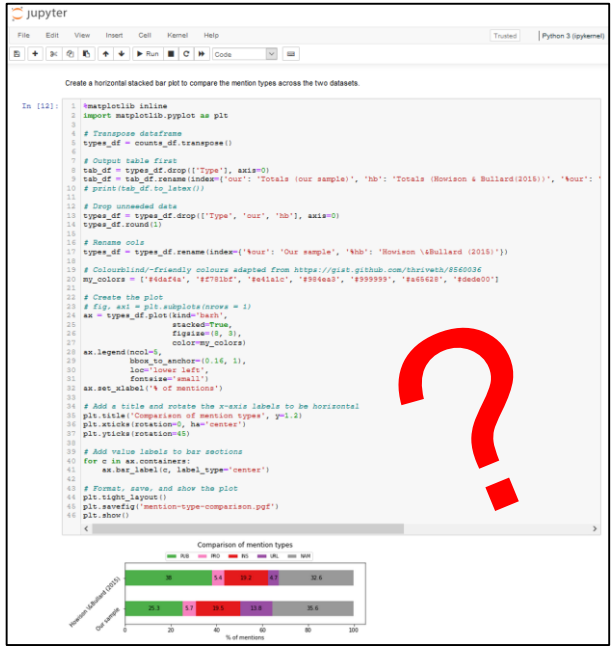
6. **Specificity**

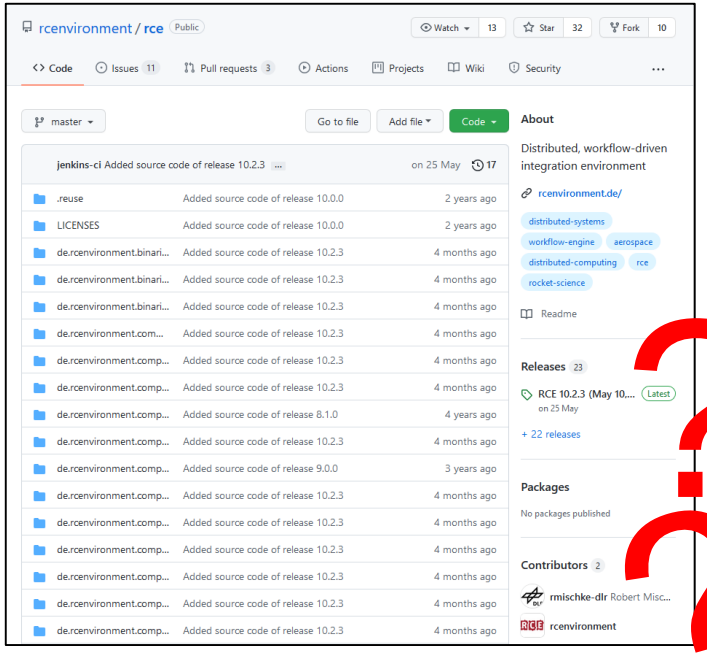
   Citation identifies the software version used in research.

Software citation metadata

- Software has **no formalized title page**
- Software name? Authors? Version? Publisher or source? Publication date?







SOFTWARE CITATION FOR DEVELOPERS AND MAINTAINERS

Bildquelle hier angeben

Making software citable

1. **Determine** complete and correct software citation **metadata**
   - Authors, name, version, publication date, identifier/locator
   - Who are the authors? Where to get a persistent identifier (PID)? Which version to tag?
2. **Provide metadata in a file** in the source code repository
   - README, LICENSE, CONTRIBUTING, **CITATION**
3. **Publish** software with metadata
4. **Maintain citation metadata** with your software

<ReSA>
Research Software Alliance
Task Force
Software Authorship & Contribution

S. Druskat, T. Krause, C. Lachenmaier, and B. Bunzeck, *Hexatomic (Version 1.4.2)*, Mar. 2023. Zenodo. DOI: 10.5281/zenodo.7778709.

Creating a CITATION file:
`CITATION.cff` – Citation File Format (CFF)

```yaml
cff-version: 1.2.0
message: If you use this software, please cite it using these metadata.
title: My Research Software
abstract: This is my awesome research software. It does many things.
authors:
  - family-names: Druskat
    given-names: Stephan
    orcid: "https://orcid.org/0000-0003-4925-7248"
version: 0.11.2
date-released: "2021-07-18"
identifiers:
  - description: This is the collection of archived snapshots of all versions of My Research Software
    type: doi
    value: "10.5281/zenodo.123456"
  - description: This is the archived snapshot of version 0.11.2 of My Research Software
    type: doi
    value: "10.5281/zenodo.123457"
license: Apache-2.0
repository-code: "https://github.com/citation-file-format/my-research-software"
```

citation-file-format.github.io

Creating a `CITATION.cff` file
Option 1: `cffinit`

citation-file-format.github.io/cffinit

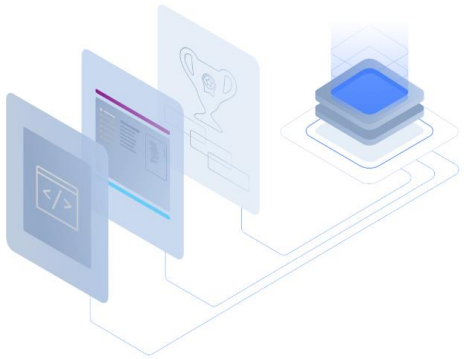
CFF INIT

Generate your citation metadata files with ease

`CITATION.cff` files are plain text files with human- and machine-readable citation information for software and datasets.

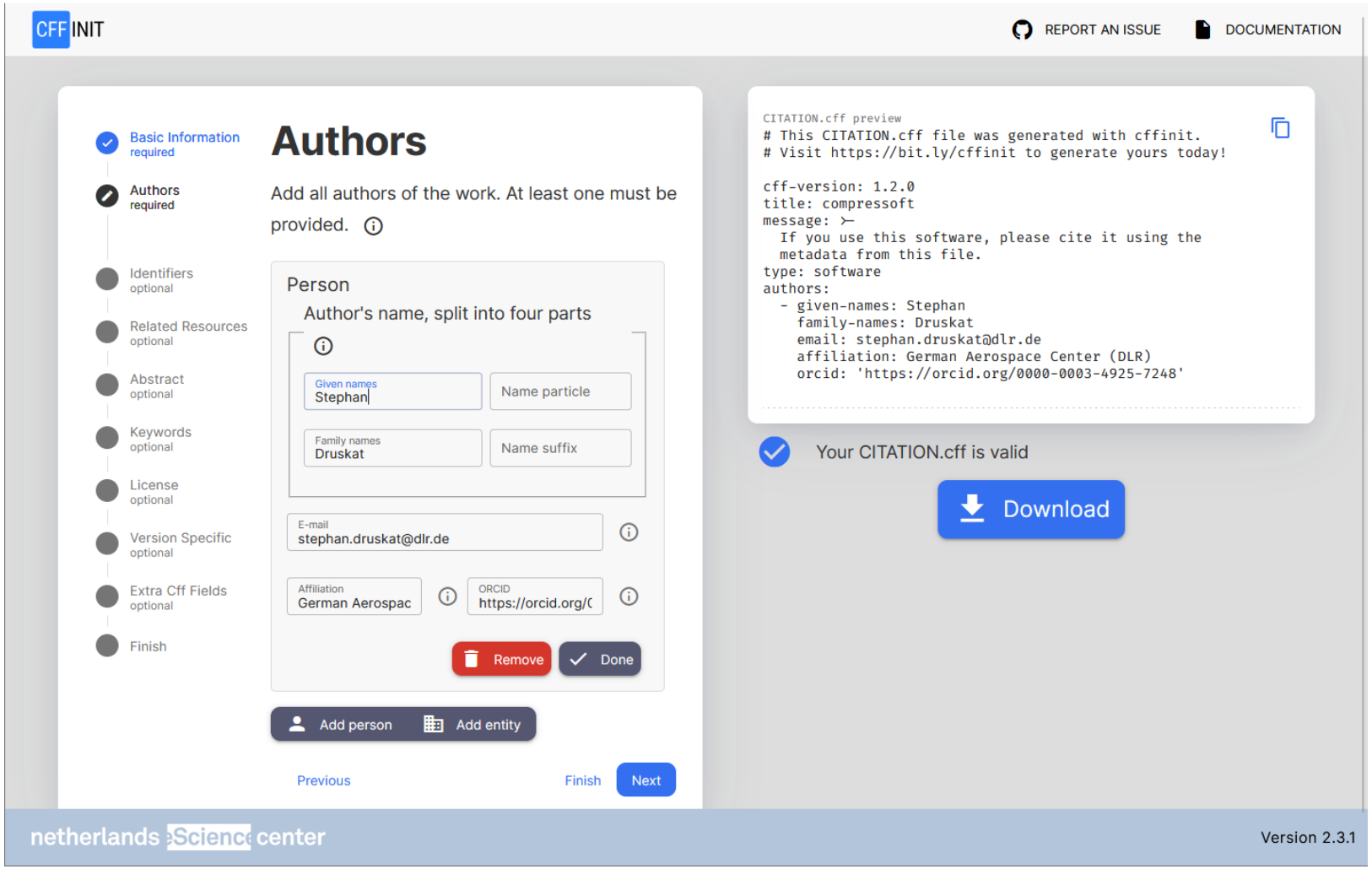
Code developers can include such files in their source code repositories to let others know how to correctly cite their software.

You can read more about the Citation File Format in the official CFF specification website.

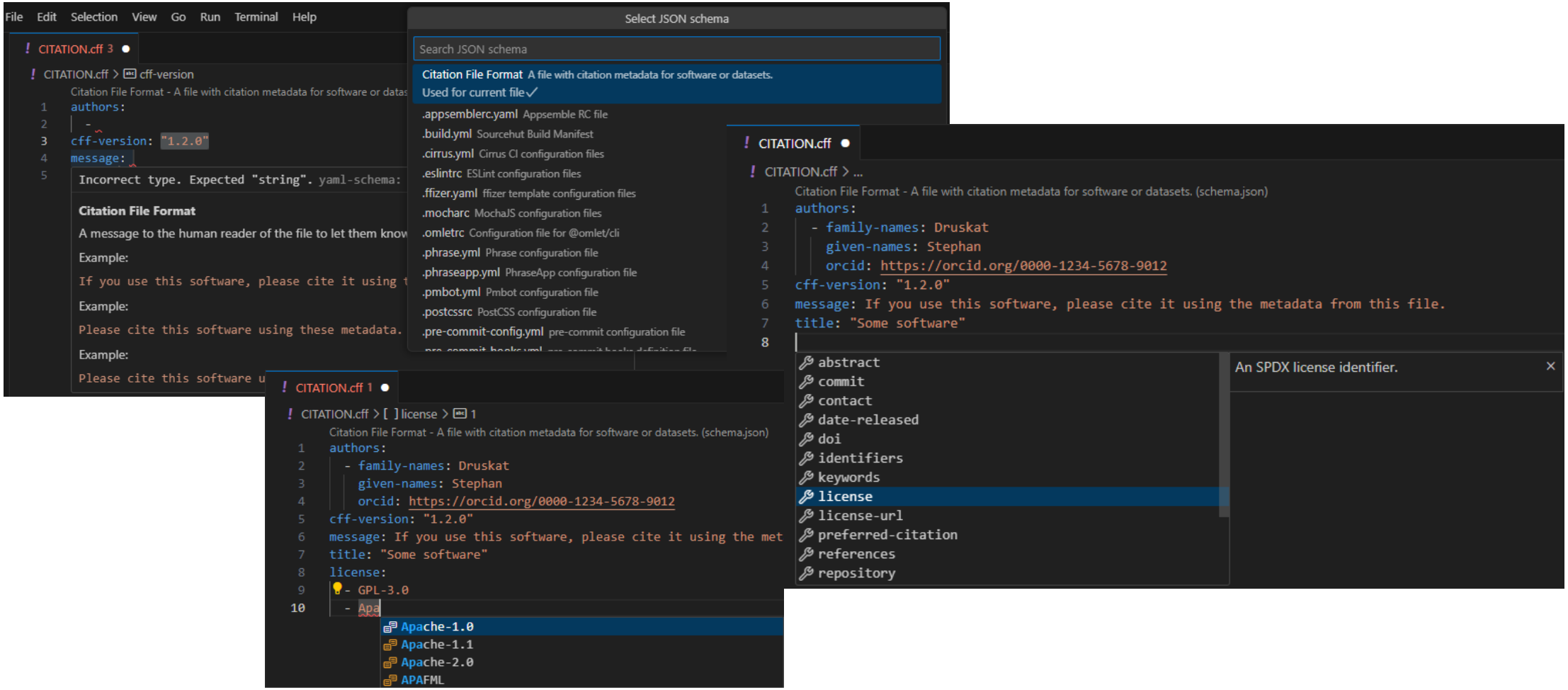


Create your CITATION.cff now, or start from an existing file!

+ Create Update



Creating a `CITATION.cff` file
Option 2: IDE integration (JSON schema , schemastore.org)



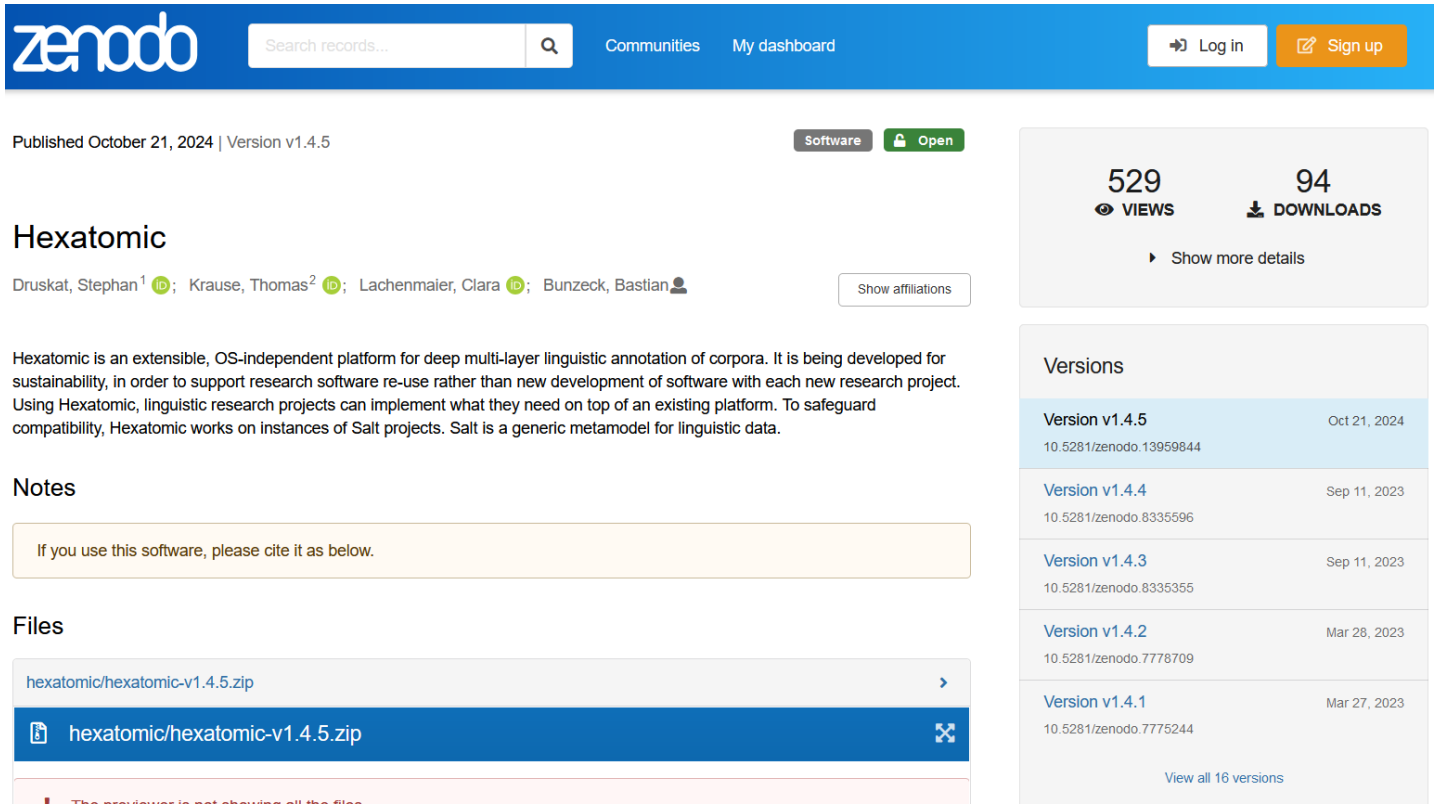
Creating a CITATION.cff file
Option 3: Copy, paste & adapt

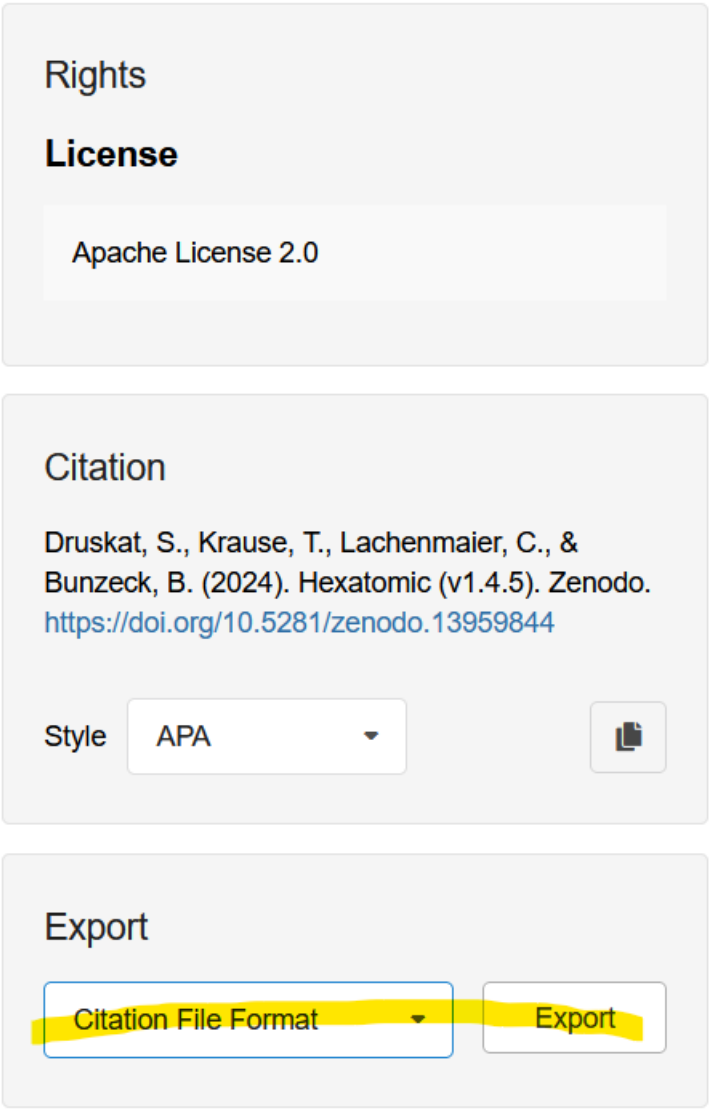
citation-file-format.github.io

- Example snippet
- Tutorials
- Schema guide:
  - Full reference
  - Snippet for typical file
  - Snippet for file with reference
  - Snippet for file with preferred citation

```yaml
cff-version: 1.2.0
message: "If you use this software, please cite it as below."
authors:
  - family-names: Druskat
    given-names: Stephan
    orcid: https://orcid.org/1234-5678-9101-1121
title: "My Research Software"
version: 2.0.4
identifiers:
  - type: doi
    value: 10.5281/zenodo.1234
date-released: 2021-08-11
```

Creating a `CITATION.cff` file
Option 4: Download from Zenodo publication

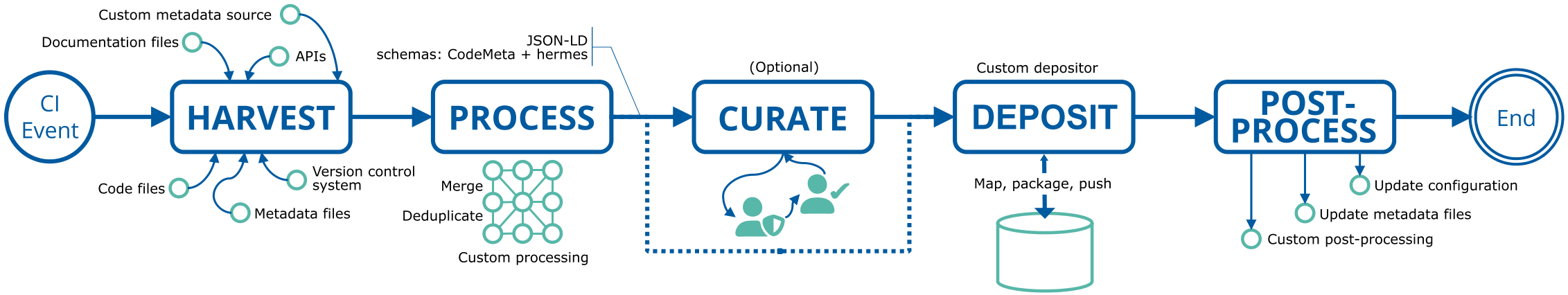




Maintaining a CITATION file

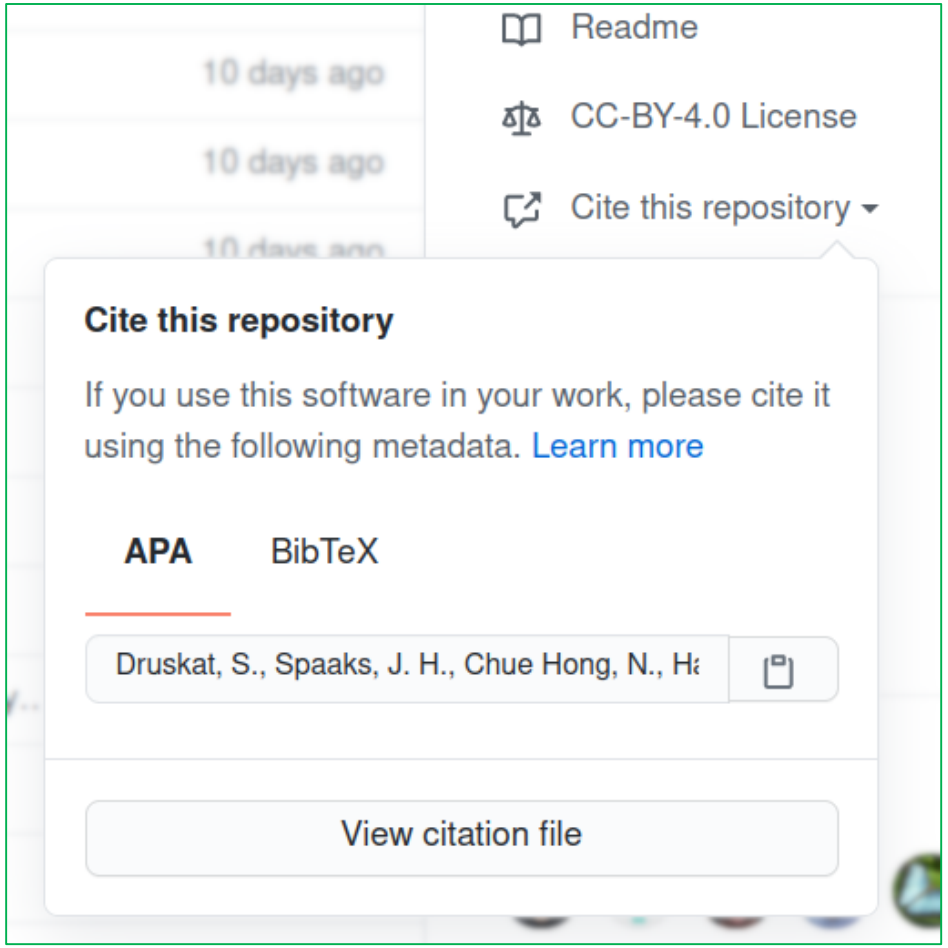
Develop citation metadata together with your software

- Validation/conversion with `cffconvert` (CLI or GitHub Action)
- Tooling available for different languages:
  - Java, JavaScript/Typescript, Julia, Python, R, Ruby, Rust, Go, Haskell, PHP, …
- HERMES develops update automation as part of publication automation

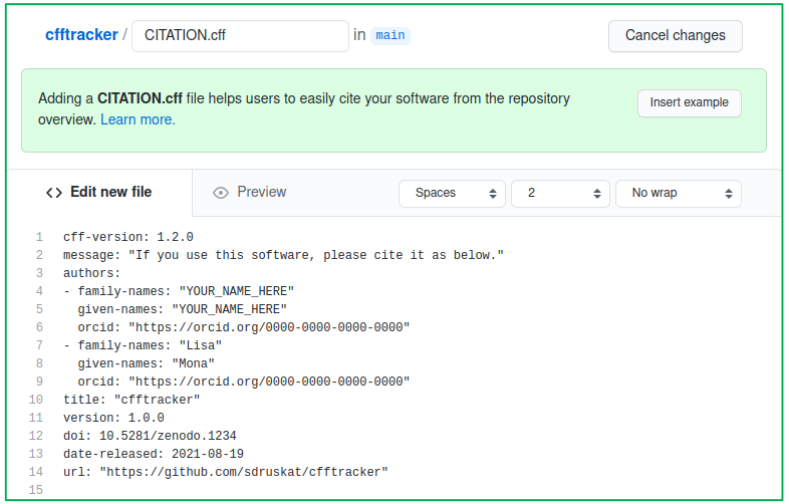


S. Kernchen, M. Meinel, S. Druskat, et al., "**Extending and applying automated HERMES software publication workflows**," doi: 10.14279/ECEASST.V83.2624.

Citation File Format integrations: GitHub*



- Detects `CITATION.cff` files
- Renders citation metadata (APA, BibTeX)
- Points to `CITATION.cff` instance
- Documents and supports creation

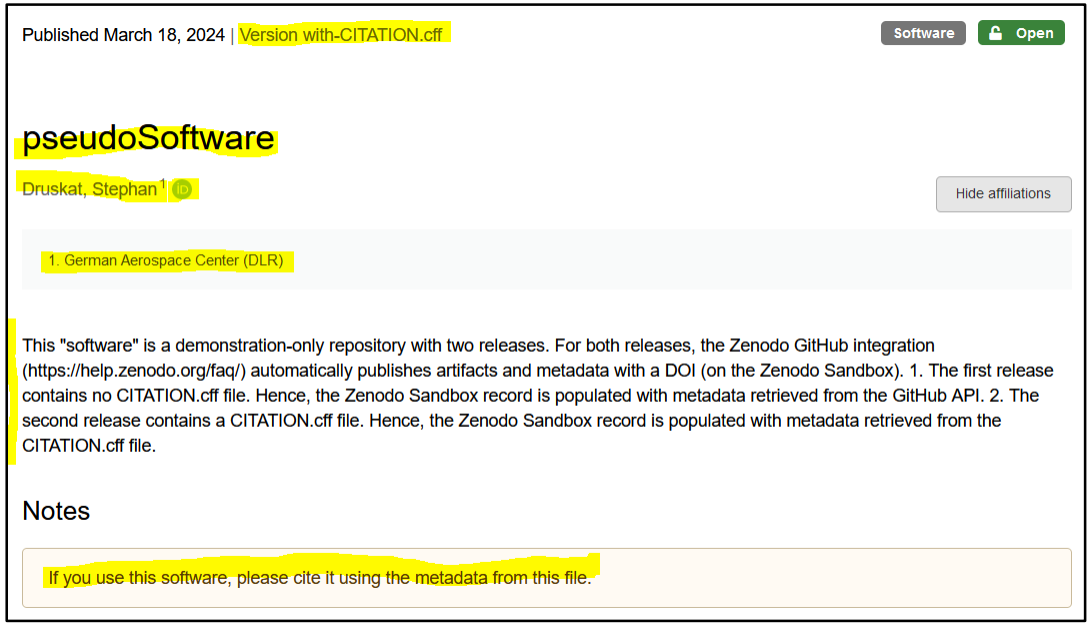


* GitLab integration under development

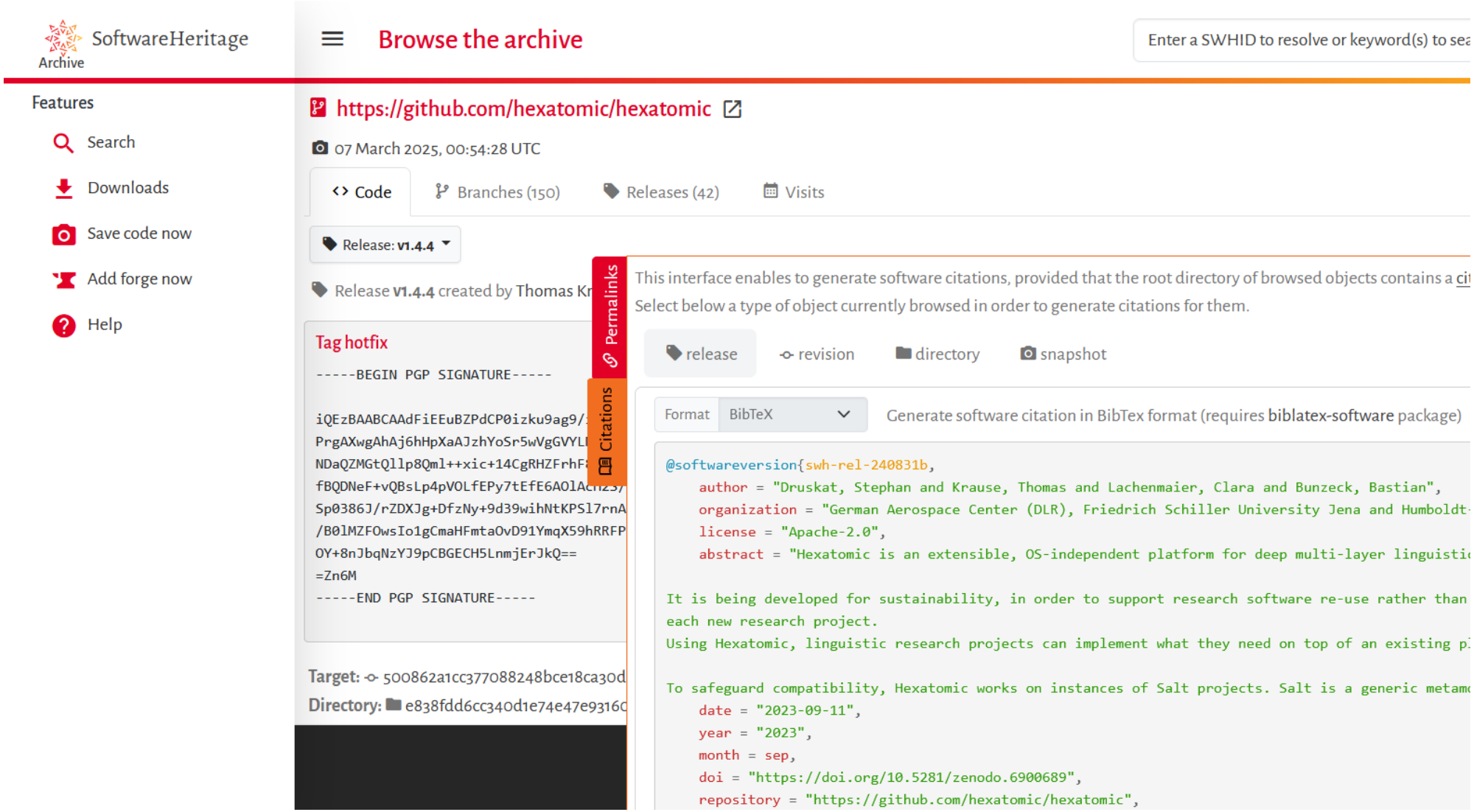
Citation File Format integrations: GitHub Zenodo integration – better metadata

- GitHub-Zenodo bridge reads metadata from `CITATION.cff`
- *Previously:* via API (GitHub release, Zenodo Sandbox deposit)
- *Now:* author-supplied metadata (GitHub release, Zenodo Sandbox deposit)

```
+ cff-version: 1.2.0
+ title: pseudoSoftware
+ message: >-
+   If you use this software, please cite it using the
+   metadata from this file.
+ type: software
+ authors:
+   - given-names: Stephan
+     family-names: Druskat
+     email: stephan.druskat@dlr.de
+     affiliation: German Aerospace Center (DLR)
+     orcid: 'https://orcid.org/0000-0003-4925-7248'
+ repository-code: 'https://github.com/citable-s0ftware/cff-github-zenodo'
+ url: 'https://citation-file-format.github.io'
+ abstract: >-
+   This "software" is a demonstration-only repository with
+   two releases.
+
+   For both releases, the Zenodo GitHub integration
```

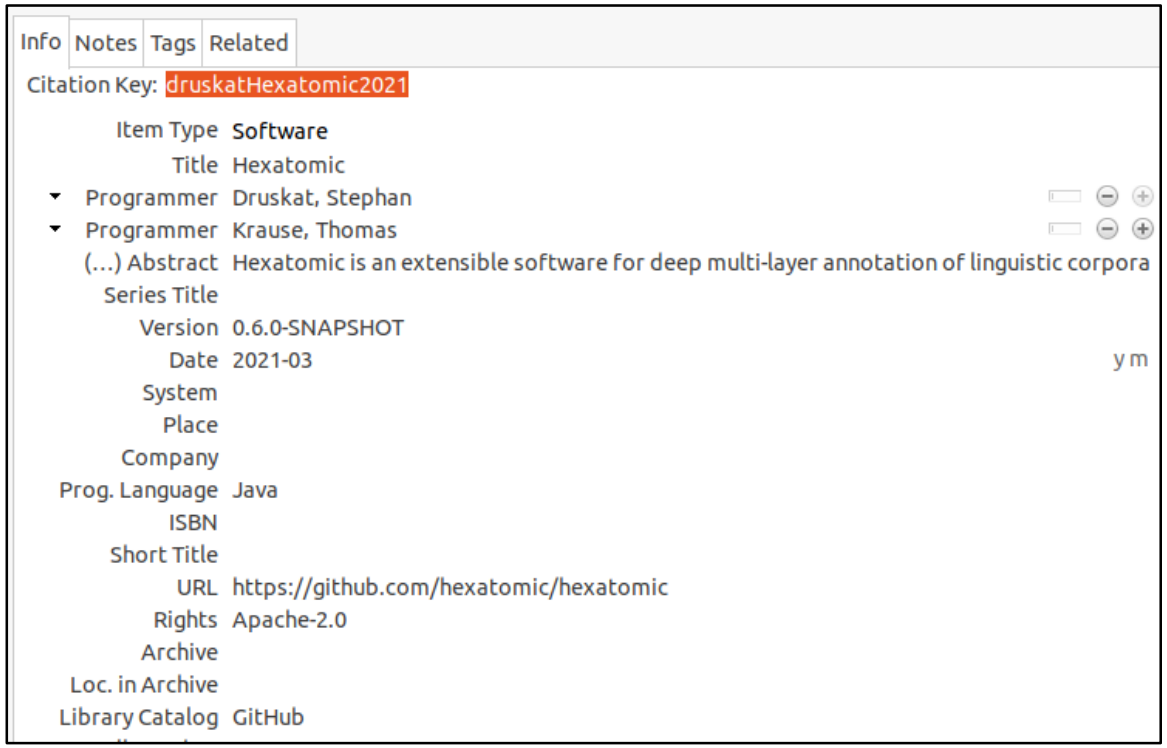


Citation File Format integrations: Software Heritage Archive

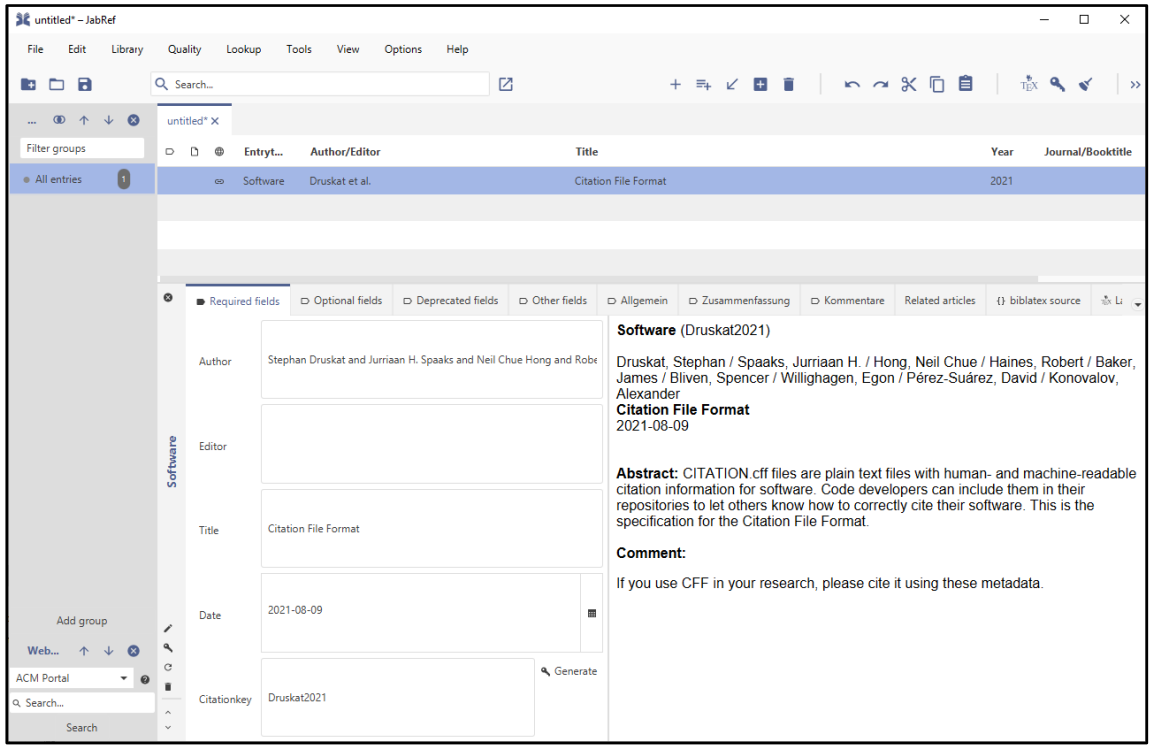


Citation File Format integrations: Reference managers

Import in [Zotero](#) from GitHub via browser plugin



Import in [JabRef](#) from `CITATION.cff` file



Publishing / archiving software versions

Software publication with a PID

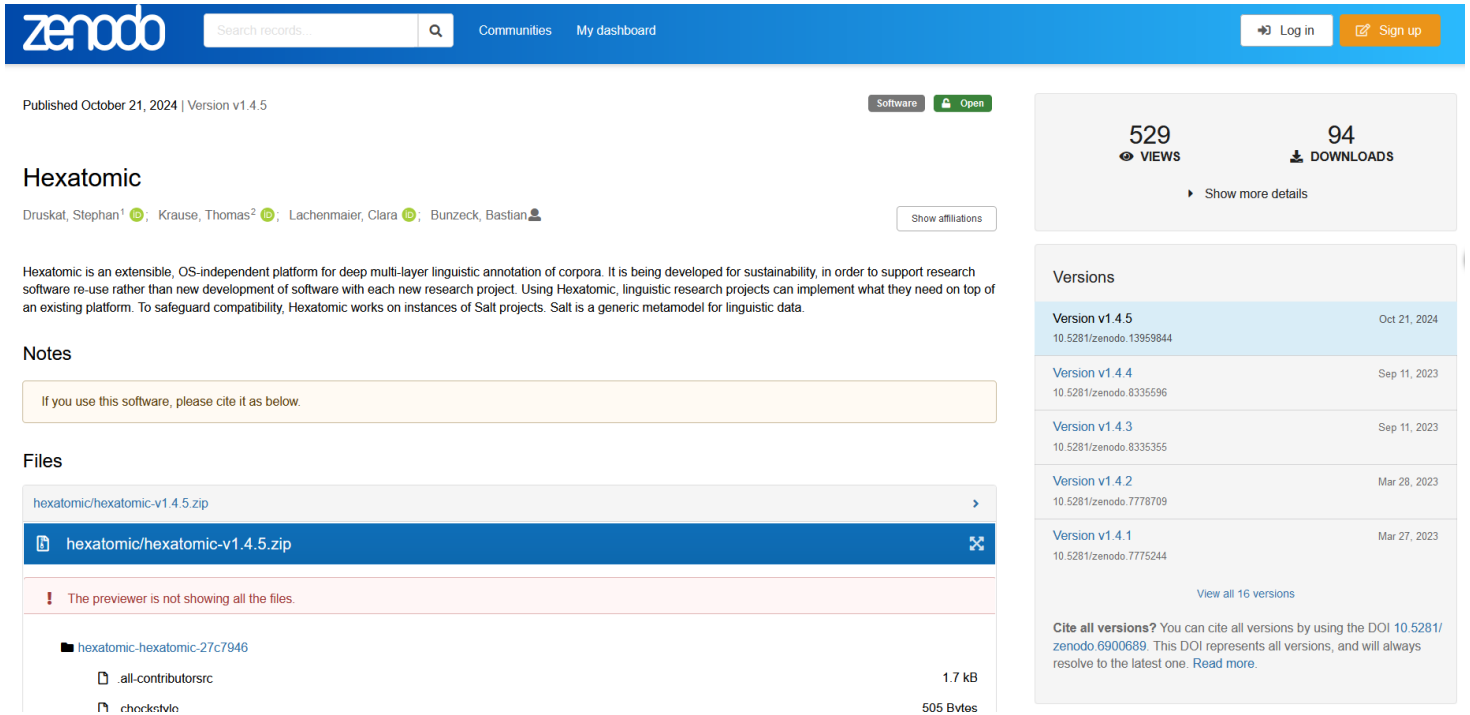
- 🥇 Institutional / discipline-specific / general purpose **repositories** (DOIs)
- 🥇 Archiving in the **Software Heritage Archive** (SWHIDs, ISO 18670)
- 🥈 Software journals (JOSS): DOI, peer review, archiving (for *one* version)

NOT software publication

- ➕ Making a package version publicly available in a package repository
- 🚫 GitHub/GitLab releases, or public repositories

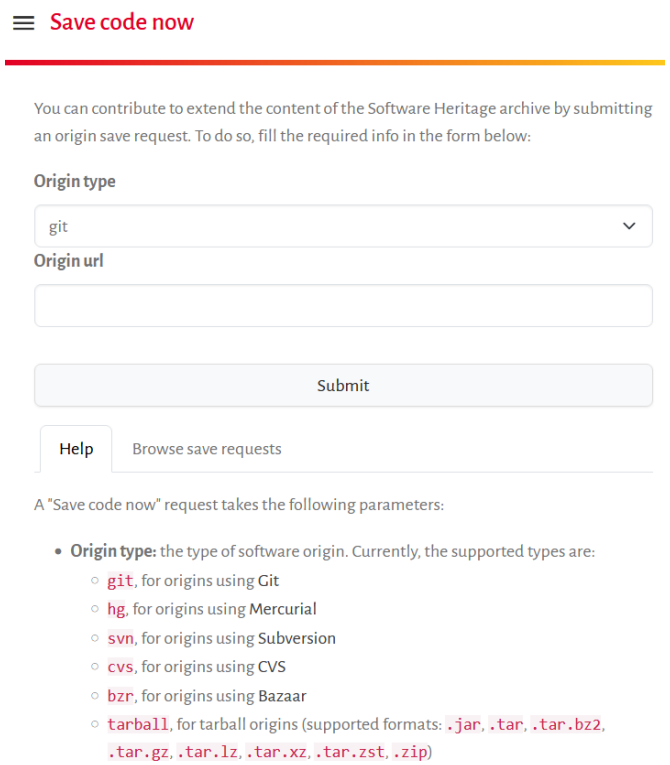
Publishing / archiving software versions: Zenodo / InvenioRDM

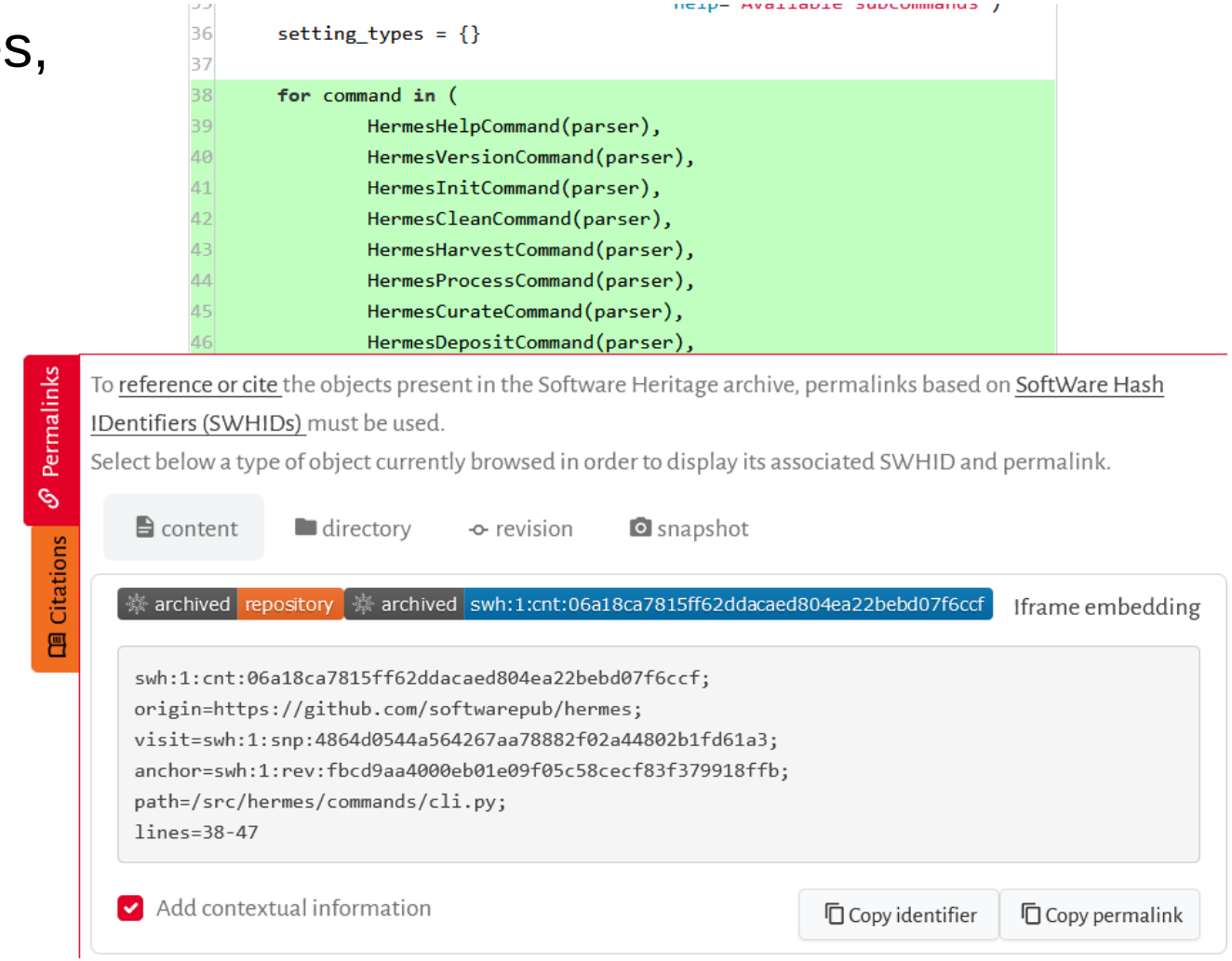
- Software type, concept/collection & version DOIs
- Manual upload, automated upload (API), DOI prereservation



Publishing / archiving software versions: Software Heritage Archive's Save Code Now

- Identifiers for revisions, releases, snapshots, directories, content





SOFTWARE CITATION FOR SOFTWARE USERS

Bildquelle hier angeben

Citing software:
Ideal and best effort cases

S. Druskat, T. Krause, C. Lachenmaier, and B. Bunzeck, *Hexatomic (Version 1.4.2)*, Mar. 2023. Zenodo. DOI: .

The hexatomic authors, *hexatomic (Version a0c59ed)*, Mar. 2023. Software Heritage Archive. SWHID: https://archive.softwareheritage.org/swh:1:dir:b9d71b12a1fcb5e7b064c27f5edecb4bcf264a2e;anchor=swh:1:rev:a0c59edf8be8c1d7c045745e4bd043f7369c7d1b.

Citing software:
When to cite software, and at which level?

Cite software whenever it represents relevant previous work

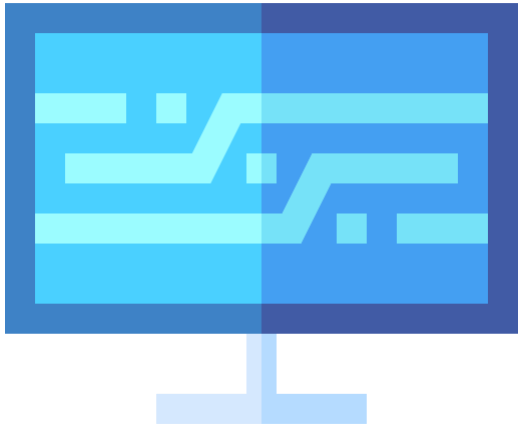
- Cite software **versions**
  - When you have used them to yield research results
  - When you discuss or compare their features or properties
  - When you use them in your own software

- Cite software **projects**
  - When you discuss properties of the project
  - When you list them as examples, e.g., for a software type

Citing software:
Which software to cite?

Cite the software that you have used, or discuss, directly

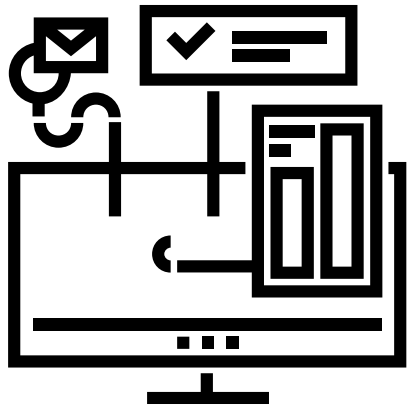
- Cite software that you have **run to process* data**
  - Models, simulations, digital twins
  - Command line tools, graphical applications, services
  - Computational notebooks, scripts
  - …
- Where relevant, cite **direct** dependencies, or **crucial** packages
- Cite software you mention or discuss



Citing software:
Cite only the software?

- Have you used the software to yield **research results**?
  → Provide **usage metadata/information** to enable reproducibility (ROCrate, computational workflows, containerization)

- Have the authors asked specifically to **cite a paper or other work**?
  → Cite the paper, but also **cite the software itself**



Citing software:
Where to get citation metadata from?

- Software distribution or documentation
- User interface (`--help`, Help > About)

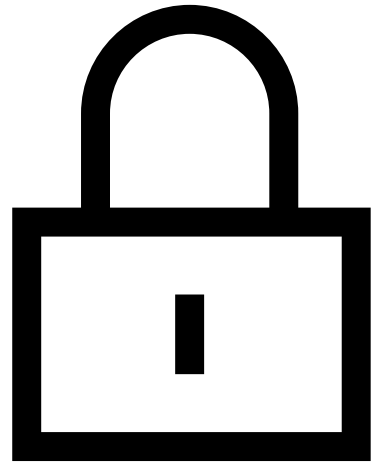
- **Open source software**
  - *Source code repository*
    - DOI or link to published version 🥇
    - Citation metadata file (CITATION.cff, codemeta.json) 🥇
    - Link to releases in package repository (PyPI, CRAN, …)?

  - *Software Heritage Archive* 🥇

Citing software:
Where to get citation metadata from?

- Software distribution or documentation

- **Closed source software**
  - User interface (`--help`, Help > About)

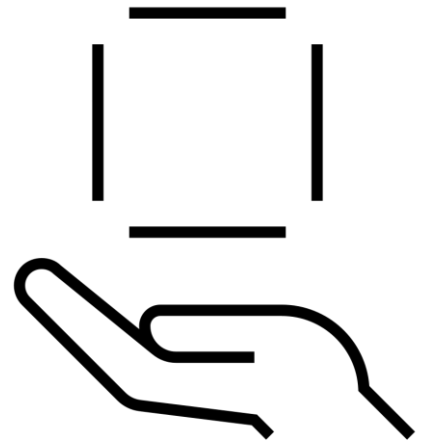


Citing software:
Which software citation metadata to trust?

Generally, DOIs and authors are the most trustworthy metadata sources

Potential pitfalls

- Publication automation (software name, authors)
- Unmaintained metadata files (authors, versions, PIDs)



⚠ **Do not use!**

- Version control / package repository metadata for authorship

Compile metadata carefully and do cross-checks. Don‘t guess metadata!

Citing software:
How to deal with missing metadata?

- **Authors‘ names**: Use a group author (*The <name> authors*)
- **Software name:** Use something that makes sense
- **Version identifier:** Use a tag/commit/revision identifier; use an MD5 checksum
- **Publication date:** Use the tag/commit/revision date
- **Platform & PID:** If possible, archive on SWH and use SWHID: <identifier>

Citing software:
How to format the software reference?

- Hack the style to include all relevant metadata
- If you use LaTeX, consider using BibLaTeX and [biblatex-software]

[2] [Software] A. Pontzen, R. Roškar, G. Stinson, and R. Woods, *pynbody: N-Body/SPH analysis for python* (Coordinated by Astrophysics Source Code Library), May 2013. ASCL: ⟨ascl:1305.002⟩,

[12] [Software Release] The CGAL Project, *The Computational Geometry Algorithms Library* version 5.0.2 (Coordinated by CGAL Editorial Board), 2020. SWHID: ⟨swh:1:rel:636541bbf6c77863908eae744610a3d91fa58855;origin=https://github.com/CGAL/cgal/⟩.

WRAP-UP

Bildquelle hier angeben

Key points

- Software citation **acknowledges the importance of software** in research, **improves the reproducibility** of research and **provides credit**
- Software citation relies on **correct and complete metadata**
- Software developers/maintainers should **provide software citation metadata**, e.g., in a `CITATION.cff` file
  - Tooling can help create and maintain citation metadata
- Software users should cite **software (versions) whenever they use** them
- Software developers should **cite the direct dependencies** of their software
- **Making software citation an established practice** requires further culture change and better tooling

Thank you!

Contact: ✉ stephan.druskat@dlr.de

💻 sdruskat.net

@sdruskat@fediscience.org

Resources: </> citation-file-format.github.io

researchsoft.org/tf-authorship-contribution

References:

A. M. Smith, D. S. Katz, K. E. Niemeyer, and FORCE11 Software Citation Working Group, "**Software citation principles**," *PeerJ Comput. Sci.*, vol. 2, no. e86, 2016, doi: 10.7717/peerj-cs.86.

S. Kernchen, M. Meinel, S. Druskat, et al., "**Extending and applying automated HERMES software publication workflows**," *Electronic Communications of the EASST*, vol. 83, 2025, doi: 10.14279/ECEASST.V83.2624.

Slides: doi: 10.5281/zenodo.15583669