WHAT RSEs SHOULD KNOW ABOUT SOFTWARE CITATION

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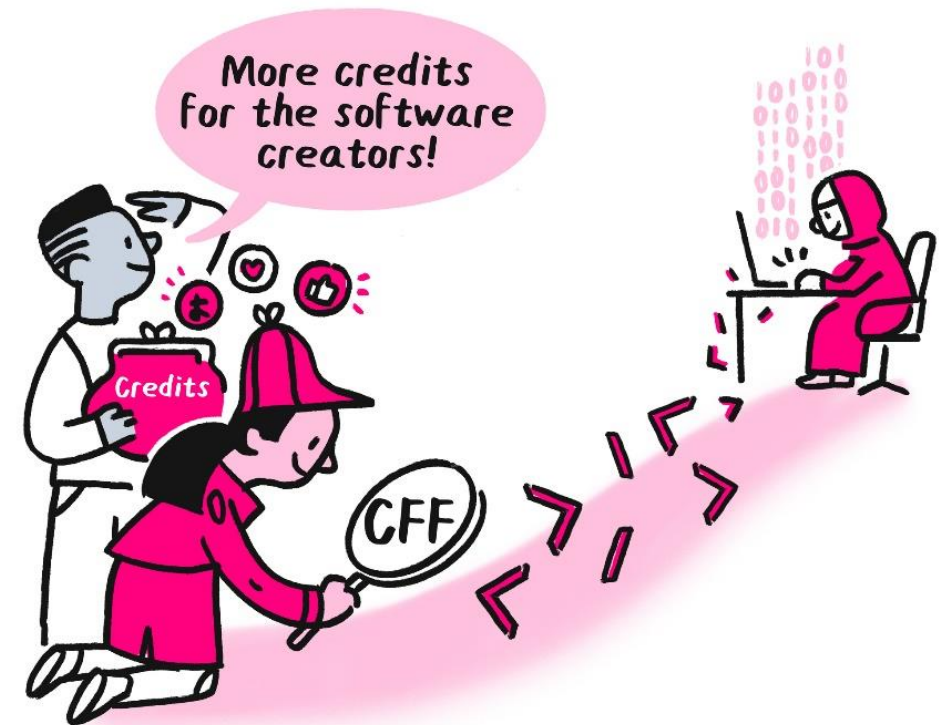
Software
Sustainability
Institute

DLR

Who / what am I?

- MA in English, modern German literature, linguistics

- RSE in linguistics (2009-2020)

- Fellow of the Software Sustainability Institute (since 2018)

- Doctoral researcher/RSE in computer science (since 2019)

- Project lead Citation File Format

- Co-founder / inaugural board member of de-RSE

- PI HERMES project (2021-2023)

Why RSEs should know about 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

How software should be cited

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.

Why software as such isn't easily citable

METADATA: Name? Authors? Version? Publisher? Publication date?

MAKING SOFTWARE CITABLE

Making software citation easier as an RSE

1. Provide citation metadata for your software!

Only you/your project knows the correct metadata:

- What's the **name** of the software?
- Who are the **authors**? [*Credit and attribution*]
  - Is the person who fixed that typo / designed the UI / wrote unit tests a "[software author](#)"?
- What is the current **version** of the software? [*Specificity*]
  - Version strings
  - What versions should people use? (Releases only? Any commit in `main`?)
- Where can people find the source code? [*Accessibility*]
- …

2. Publish your software!

Making software citation easier as an RSE in practice:
Provide citation metadata with the Citation File Format (CFF)

citation-file-format.github.io

```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"
```
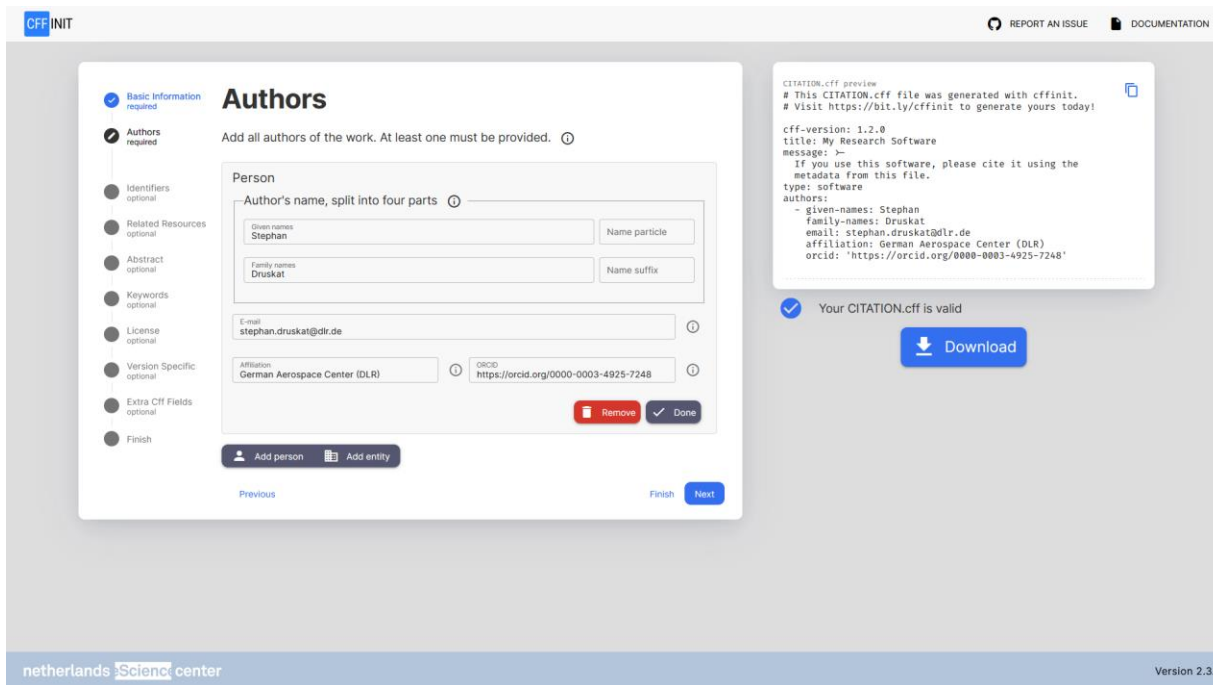
+ references, preferred citation, experimental dataset support

METADATA: In a `CITATION.cff` file in the source code repository

Why use CFF?
1. Tooling: Create, edit & update, validate, convert

cffinit: web tool for CFF file creation - https://bit.ly/cffinit

Overview: https://bit.ly/cff-tools-list

Why use CFF?
2. GitHub support

DLR

- Detects `CITATION.cff` files

- Renders citation metadata (APA, BibTeX)

- Points to `CITATION.cff` instance

- [Documents](#) and supports creation

Why use CFF?
3. Zenodo support

Why use CFF?
4. Reference manager support

Import in **Zotero** from GitHub via browser plugin

Import in **JabRef** from `CITATION.cff` file

Why use CFF?
5. IDE support

■ Auto-completion and validation via JSON Schema / schemastore.org

Why use CFF?
6. Everyone does it

- GitHub (if the GitHub Search API is right) has **~15,000** CFF files

Getting started with the Citation File Format

Validity of v1.2.0 CFF files created with/without cffinit
(preliminary data)
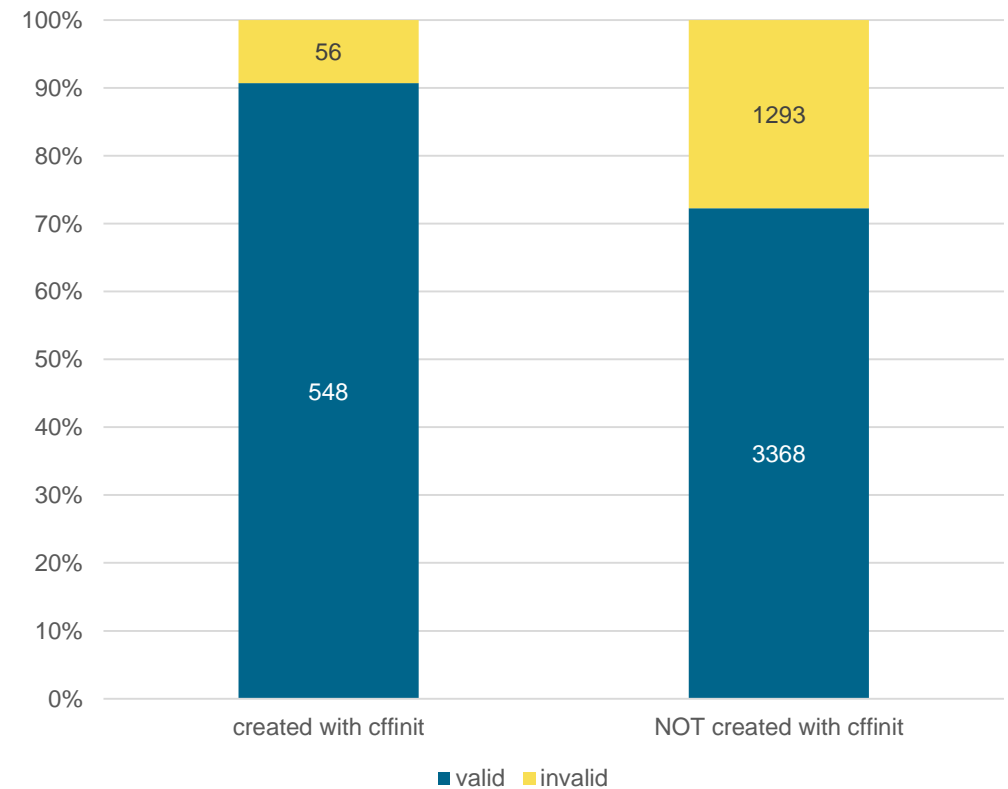
1. Use https://bit.ly/cffinit to create an initial CFF file for your software!
2. Optionally, refine the file using IDE support.

PUBLISHING SOFTWARE

Why publish software?

- **To enable software citation**

- To make the software FAIR

  - Findable, accessible, (interoperable, reusable*)

- To let software count towards evaluation
  (key performance indicators, etc.)

  - Based on Helmholtz Open Science Policy, number of citable software publications are counted (from 2023 onwards)

Publishing software (for software citation)

- State of the art: publish software (or at least citation metadata) to an (ideally open access) **publication repository** 🥇
  - Repository publication > software journals
  - The source code repository is not a publication repository!

In practice:

- Publish every release on **Zenodo** (or your institutional repository).
- This gives you:
  - A DOI [*Unique identification*]
  - An archived snapshot [*Persistence*]
  - A resolvable identifier whose target provides access to the software [*Accessibility*]
  - An identifier that points to the specific version [*Specificity*]

Making software publication easier for RSEs: CFF and the GitHub Zenodo integration

Making software publication easier for RSEs: Automation with HERMES

- Automating software publication via CI

- Harvest and merge existing metadata

- Pushing proactively to keep full control

- Curation & FAIR metadata-only publication (closed source) possible

- InvenioRDM and Dataverse support
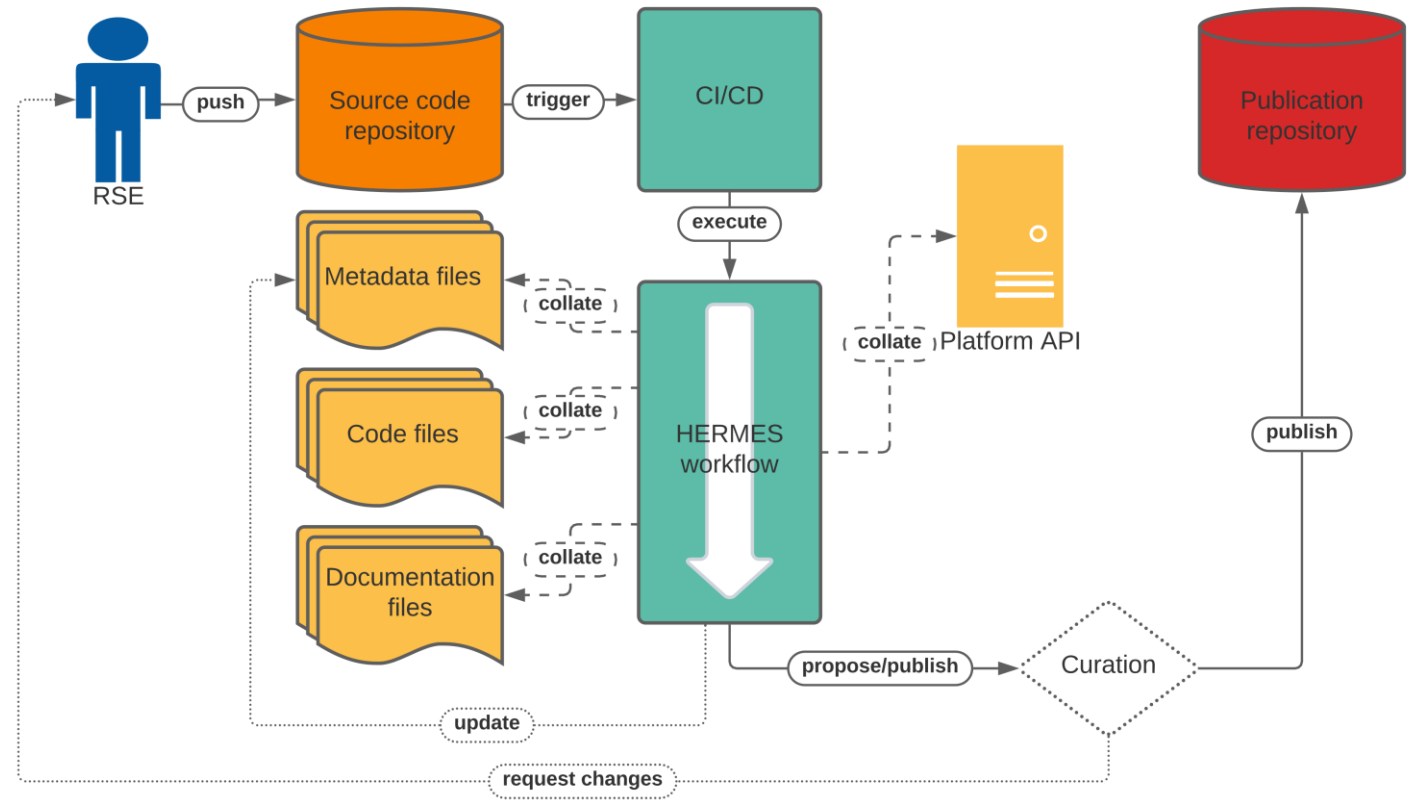
HERMES workflow, high-level view, HERMES project (CC BY-SA 4.0)

Making software publication easier for RSEs: Automation with HERMES

software-metadata.pub

- Helmholtz Metadata Collaboration project (DLR + FZJ + HZDR)

- **deRSE24** workshop:
  S. Kernchen: *Automating your FAIR software publications with HERMES*

Conclusion

1. **(Enabling) software citation is part of our job as RSEs.**
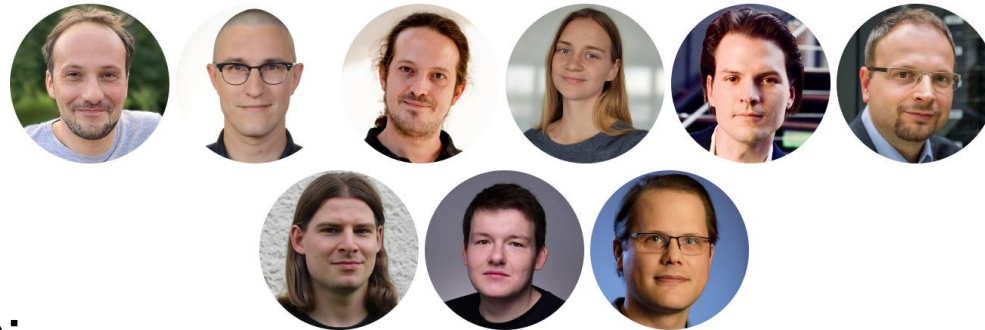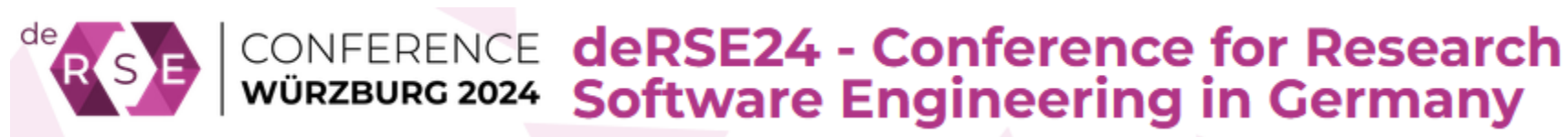
2. Software creators/RSEs should provide and maintainer the **correct citation metadata** for software in a Citation File Format `CITATION.cff` file in the source code repository.

3. Software creators/RSEs should **publish every release** to Zenodo (or another publication repository).

4. Researchers **must cite software** as they would cite a paper.

Thanks!

The **Citation File Format** has been funded by the *German Aerospace Center (DLR)*, the *Netherlands eScience Center*, the *Software Sustainability Institute*, *Code for Science & Society*, and *CampusSOURCE e.V.*

Projects

https://citation-file-format.github.io/ | https://software-metadata.pub

Contact

stephan.druskat@dlr.de | Fediverse: @sdruskat@fosstodon.org

ORCID: https://orcid.org/0000-0003-4925-7248