AUTOMATING RESEARCH SOFTWARE PUBLICATION WITH RICH METADATA

What can we learn for research data publication?

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

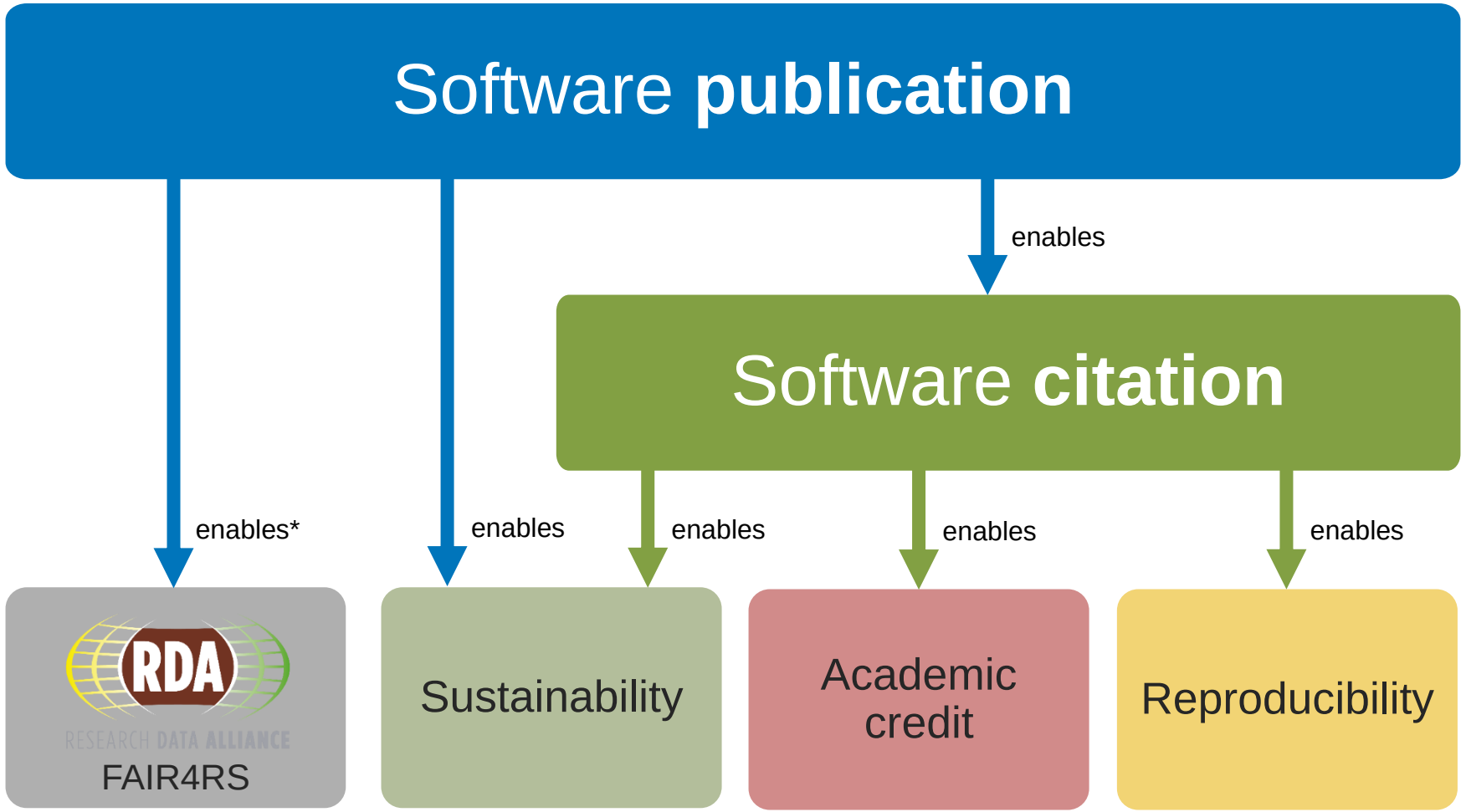
DOI 10.5281/zenodo.14265448

DLR

Overview

- Software publication & FAIR research software
- HERMES concept
- HERMES implementation
- Current & future work
- Automating research *data* publication with rich metadata?
- [optional] Example

Software *publication* is important, because research software is important!



Adapted from Druskat et al. (2023)

FAIR for Research Software (FAIR4RS)

- **F:** Software, and its associated metadata, is **easy for both humans and machines to find**.
- **A:** Software, and its metadata, is **retrievable via standardized protocols**.
- **I:** Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.
- **R:** Software is both usable (can be executed) and reusable (can be **understood**, modified, built upon, or incorporated into other software).

Chue Hong et al. (2022)

Publication of software with metadata
(Research) Software Engineering

What is software publication?

Publication of metadata and artifacts* for software **versions** in publication repositories

Persistent identifier for each version

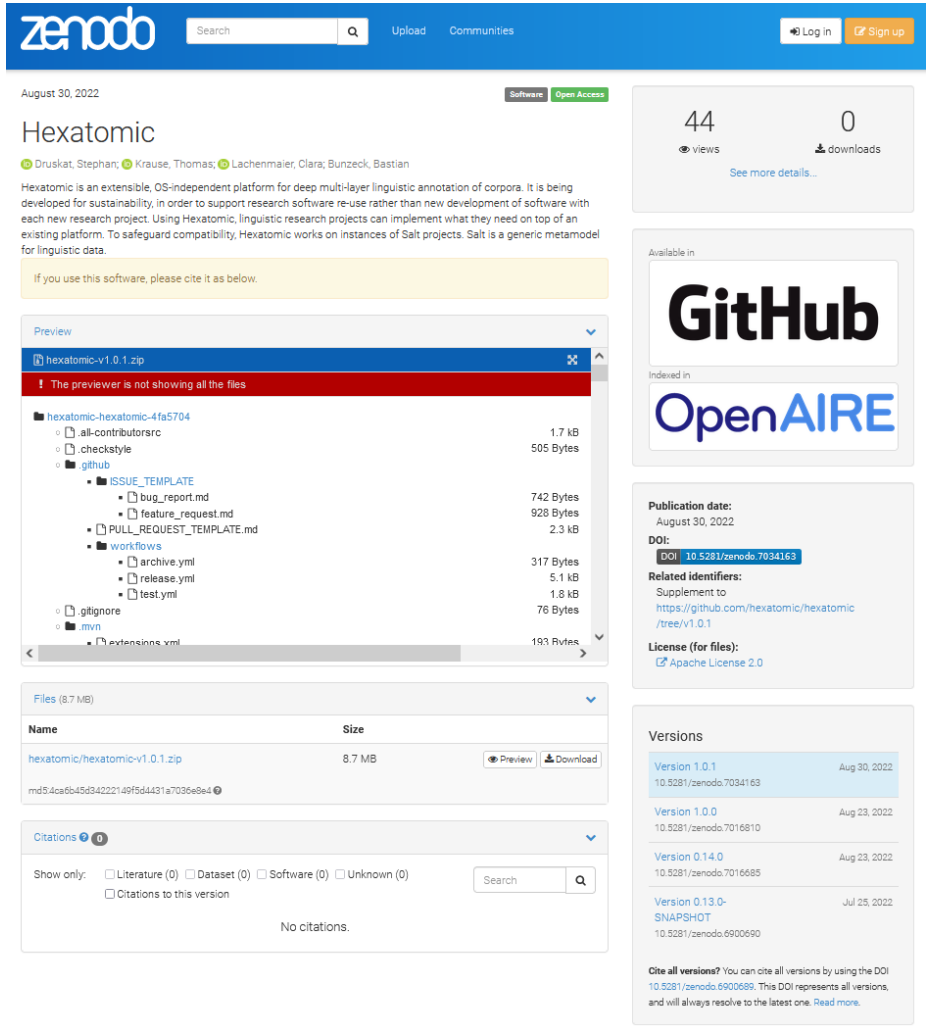
Not software publication:

- Software available on a source code platform (*GitHub*, *GitLab*, etc.)
- A paper *about* the software

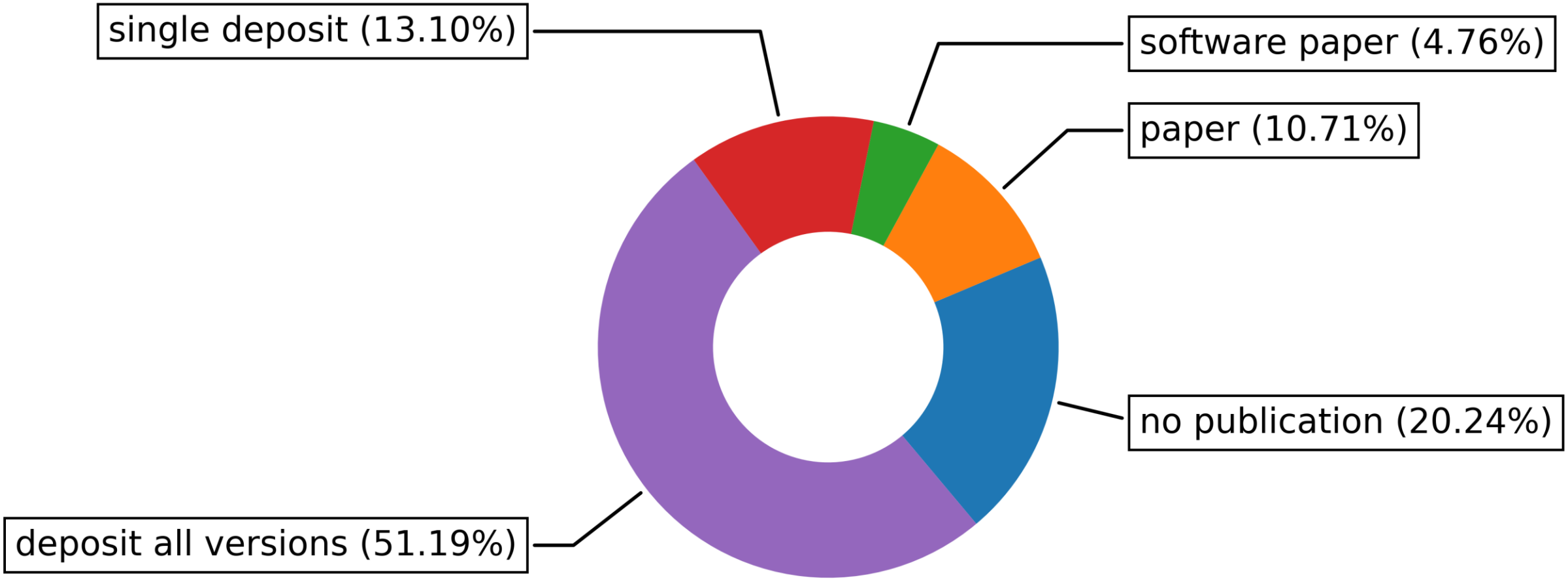
Interim solution:

- software journals (JOSS, JORS, etc.)

Challenge: quality assurance

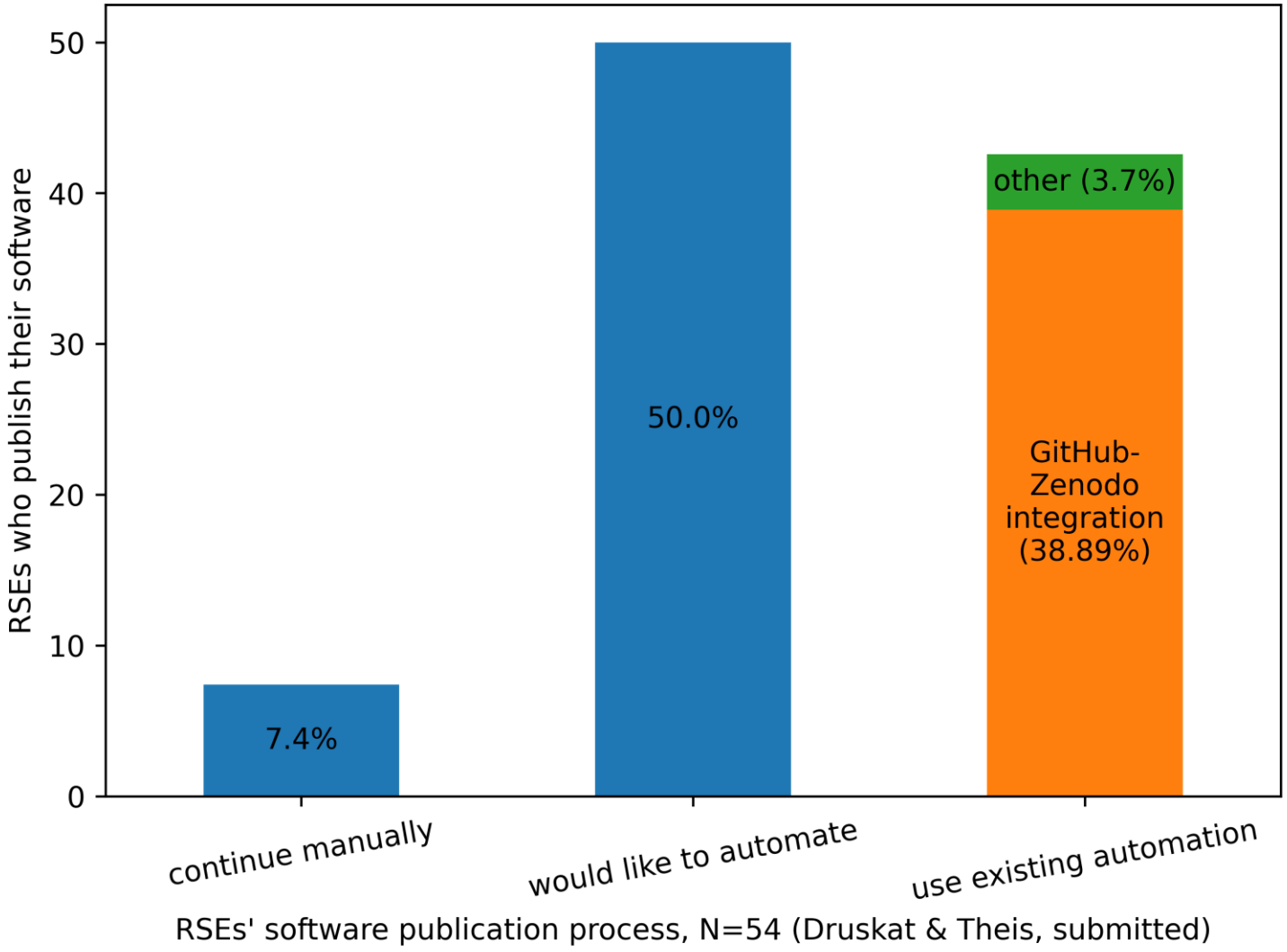


How is software currently published/publicized?



RSE software publication practices, N=84 (Druskat & Theis (submitted))

How do RSEs publish their software?



Automated software publication: Zenodo-GitHub integration

DOIs for versions & projects, metadata, persistence

GitHub-Zenodo integration,
improved metadata via `CITATION.cff`

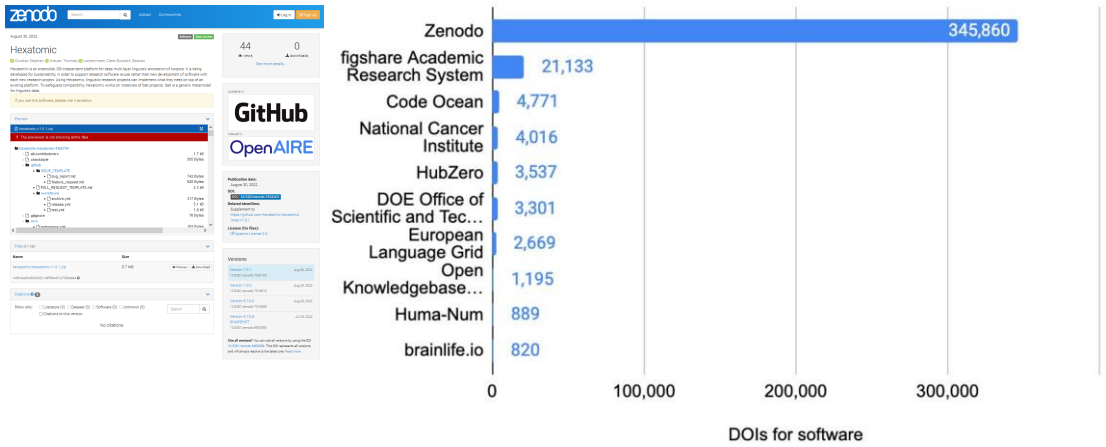
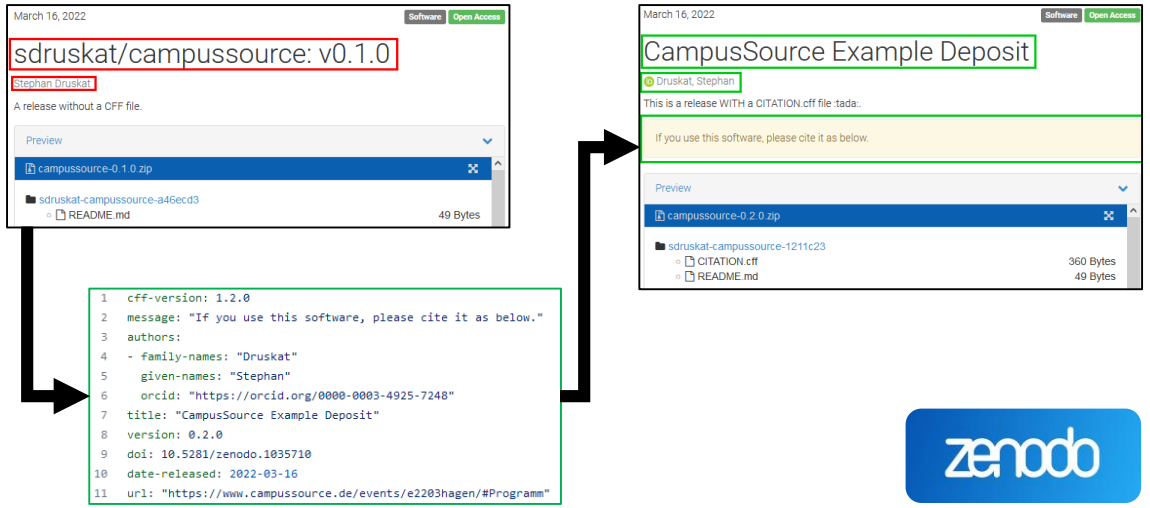


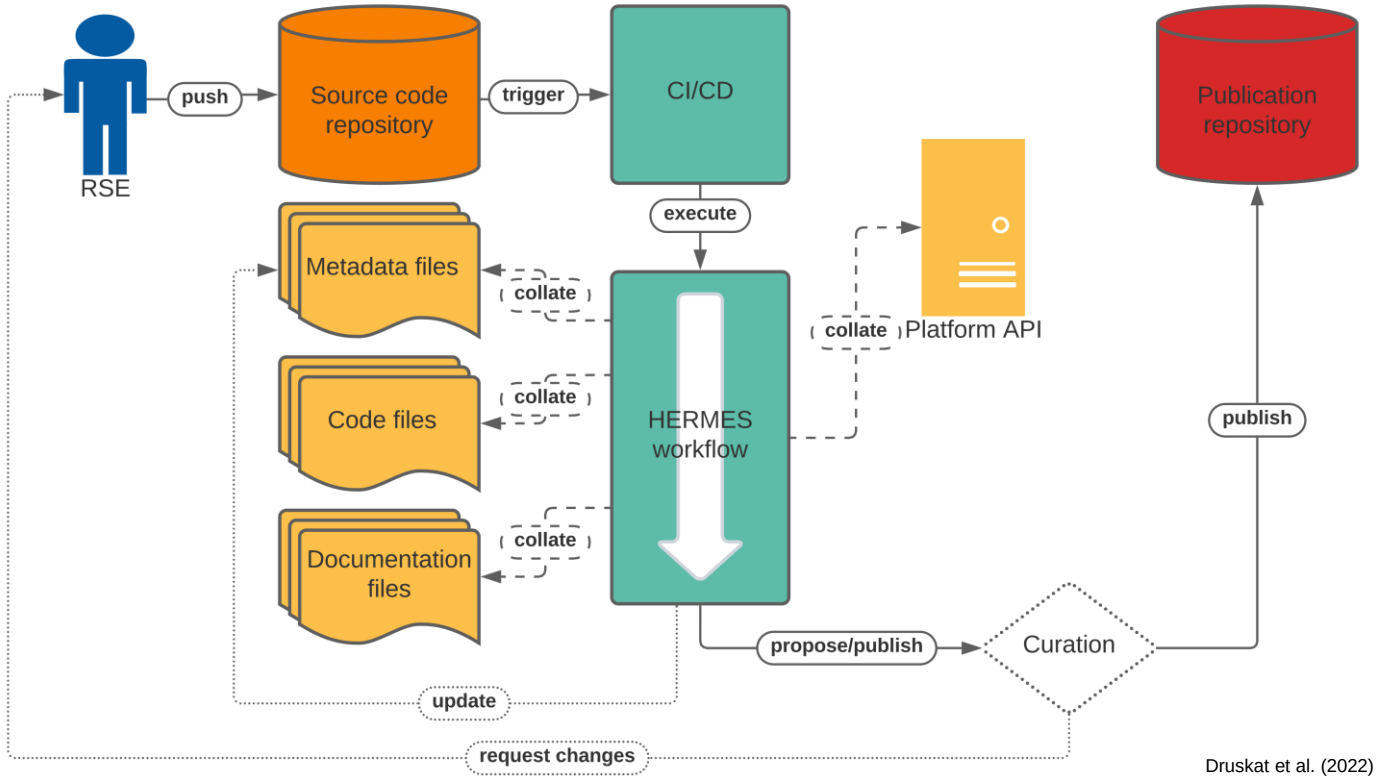
Figure from Mejias (2023).



- Most often used by far: Zenodo (developed & run at CERN), GitHub integration
- Persisted records, persistent identifiers (DataCite DOIs w/ metadata), landing pages
- Integration only for specific platform combination, insufficient metadata in basic usage

HERMES: Helmholtz Rich Metadata Software Publication
(Helmholtz Metadata Collaboration project)

- Automated software publication for all platform combinations
- Use existing metadata to enrich records/improve FAIRness
- Enable:
  - closed source publication,
  - curation & sign-off processes,
  - updating metadata records

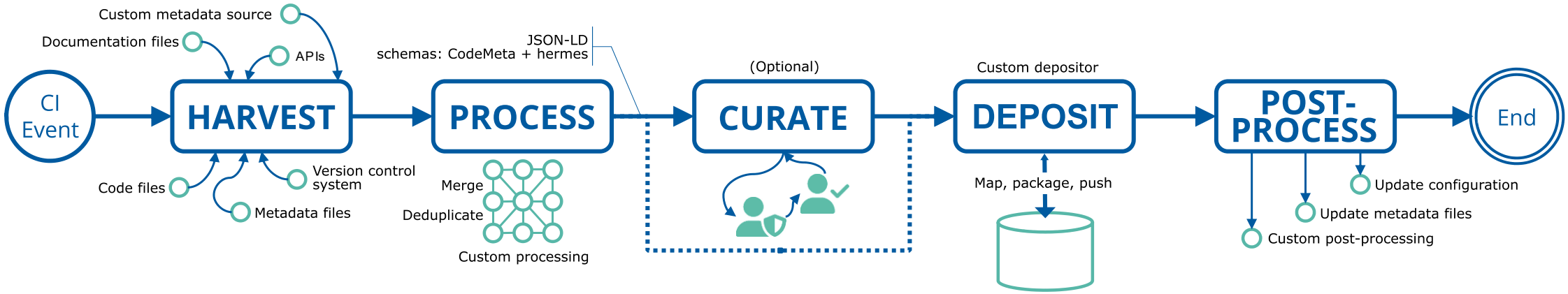


Druskat et al. (2022)

- software-metadata.pub

HERMES: Implementation

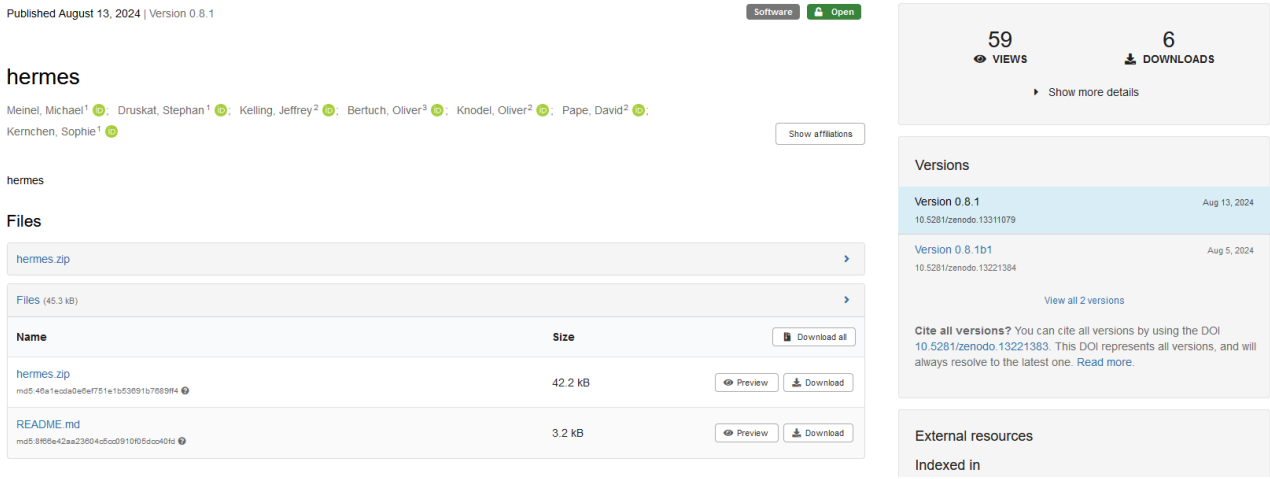
- Continuous integration workflow: on <event> run `hermes` as configured
- Tutorials for GitHub/GitLab: docs.software-metadata.pub



- `hermes` Python package (Meinel et al. 2024) + CI templates (GitHub, GitLab)
- Plugins via Python Extension Point mechanism for each step
- Details: Kernchen et al. (2024)

HERMES: Use and extensibility

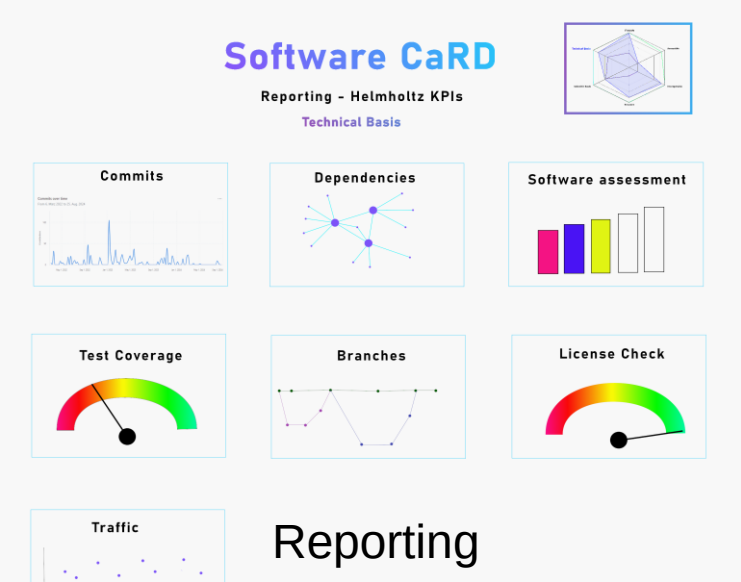
- Dogfooding:
  - https://github.com/softwarepub/hermes/releases/tag/v0.8.1
  - https://zenodo.org/records/13311079
- User testing at live coding workshop (deRSE24)
- `hermes-plugin-python` (Fritzsche & Meinel 2024) created by high school intern within a few days
- Details: Kernchen et al. (2024)



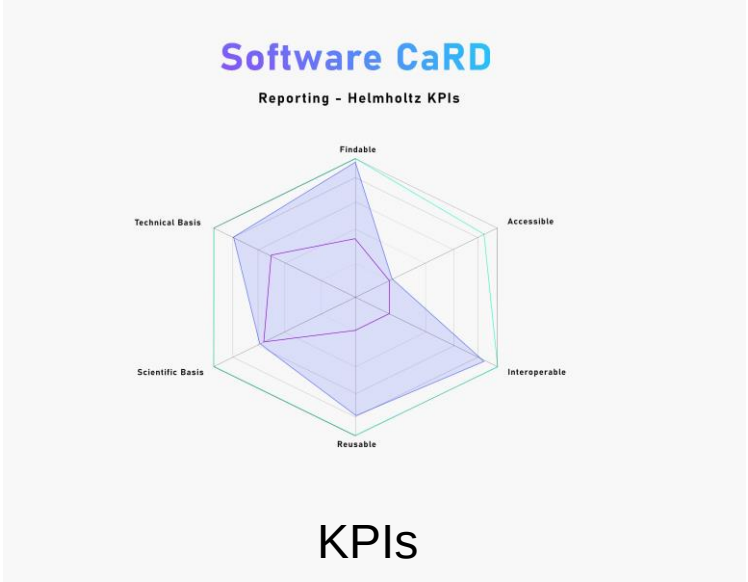
Current work: hermes 1.0.0, Software CaRD

- hermes 1.0.0
- Software Curation and Reporting Dashboard (Software CaRD)
  - Input: Consistent knowledge graph produced by HERMES
  - Compliance checks against configurable policies (KPIs, curation)

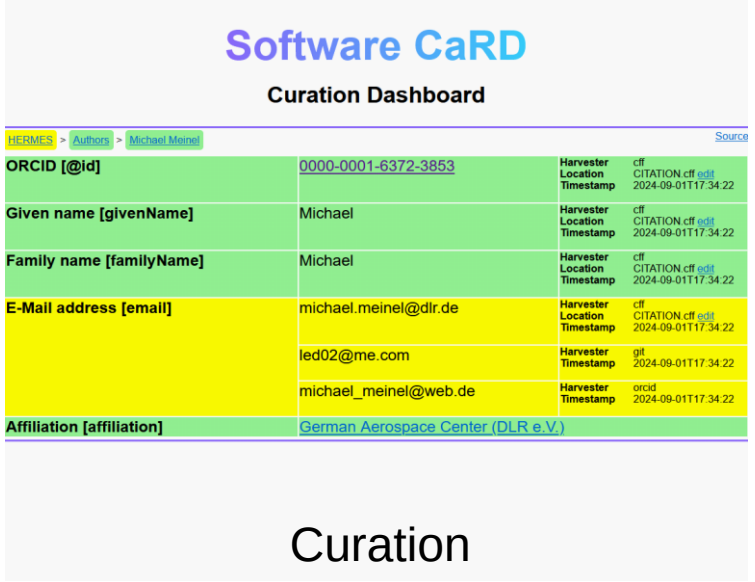

```
{
"@context": [...],
"@type": "SoftwareSourceCode",
"name": "hermes",
"version": "0.8.1",
"license": "https://spdx.org/licenses/Apache-2.0",
"author": [
{
"@id": "https://orcid.org/0000-0001-6372-3853",
"@type": "Person",
"affiliation": {...},
"familyName": "Meinel",
"givenName": "Michael",
"email": "michael.meinel@dlr.de"
},
...
],
"hasPart": [
{
"@type": "CreativeWork",
"name": "README",
"encoding": {
"@type": "TextObject",
"encodingFormat": "text/markdown",
"url": "file://./README.md"
}
},
...
]
}
```

Reporting



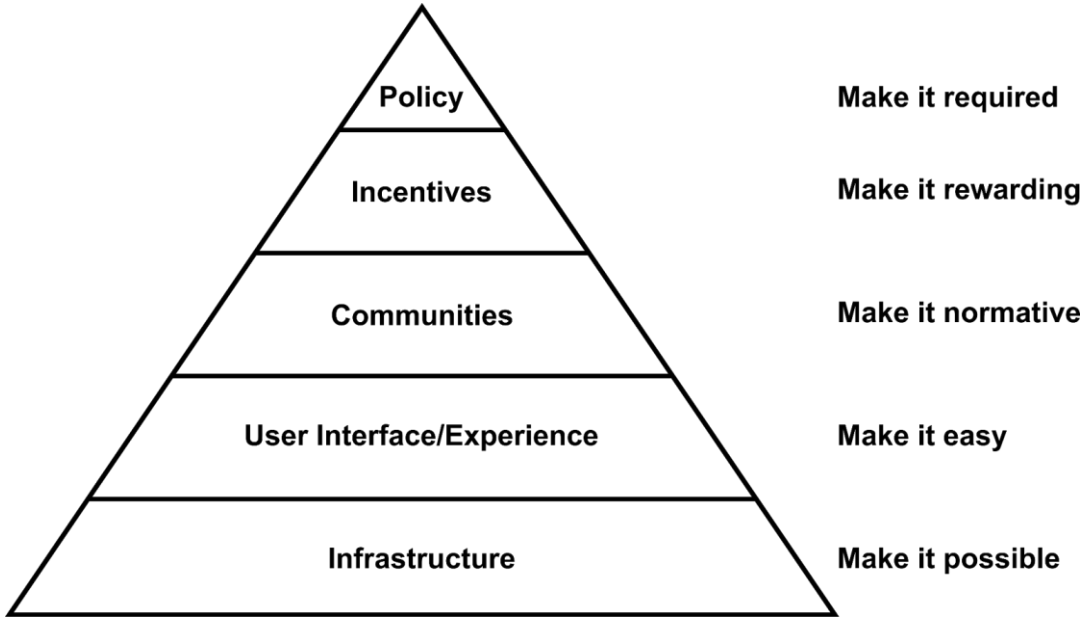
KPIs



Curation

Future work

- Rollout of HERMES workflows at two German universities
  - Heterogeneous stakeholders
  - Empirical requirements elicitation & usability optimization
  - New deployment targets: DSpace, MyCoRe
  - Development of targeted capacity building & outreach material



Automating research *data* publication with rich metadata?

- Data ≠ software
  - Executability
  - Evidence vs. tool
  - Creative work vs. facts/observations
  - Change cycles & dependencies
- Data ≠ software?
- Do we have the same problems in data as in software?
  - Dynamicity
  - Workflows
  - Infrastructure

Practicalities of using HERMES for data

- Clear focus on research *software*
- HERMES data model: CodeMeta++ (v3.0, Jones et al., 2023)
- Linked Open Data
- Interaction through `HermesContext` API with internal data (nested `dict`s)

Conclusion

- HERMES automates the publication of software with rich metadata
- HERMES supports FAIR software as FAR as possible
- Can HERMES be used to publish FAIR research datasets?
  - Yes, but it would take some work
- Should HERMES be used to publish FAIR datasets?
  - It depends …

Thanks!

HERMES: **software-metadata.pub**, **docs.software-metadata.pub**
github.com/softwarepub/hermes, **github.com/softwarepub/ci-templates**
pypi.org/project/hermes/, HMC project ZT-I-PF-3-006

Me: Stephan Druskat, stephan.druskat@dlr.de, fosstodon.org/@sdruskat

References:

Druskat, S., Bertuch, O., & Struck, A. (2023). Towards Research Software-ready Libraries. ABI Technik, 43(3), 168-178. doi: 10.1515/abitech-2023-0031.

Chue Hong, N P., et al. (2022). FAIR Principles for Research Software (FAIR4RS Principles). RDA. doi: 10.15497/RDA00068.

Druskat, S., & Theis, S., (submitted). Challenges in designing research infrastructure software in multi-stakeholder contexts, in 27th International Conference on Human-Computer Interaction, Gothenburg, Sweden, submitted.

Mejias, G., (2023). DOIs for research software: Increasing Visibility, Connectivity, Citability. Zenodo. doi: 10.5281/zenodo.7985199.

Druskat, S., Bertuch, O., Juckeland, G., Knodel, O., & Schlauch, T., (2022). Software publications with rich metadata: State of the art, automated workflows and HERMES concept. arXiv. doi: 10.48550/arXiv.2201.09015.

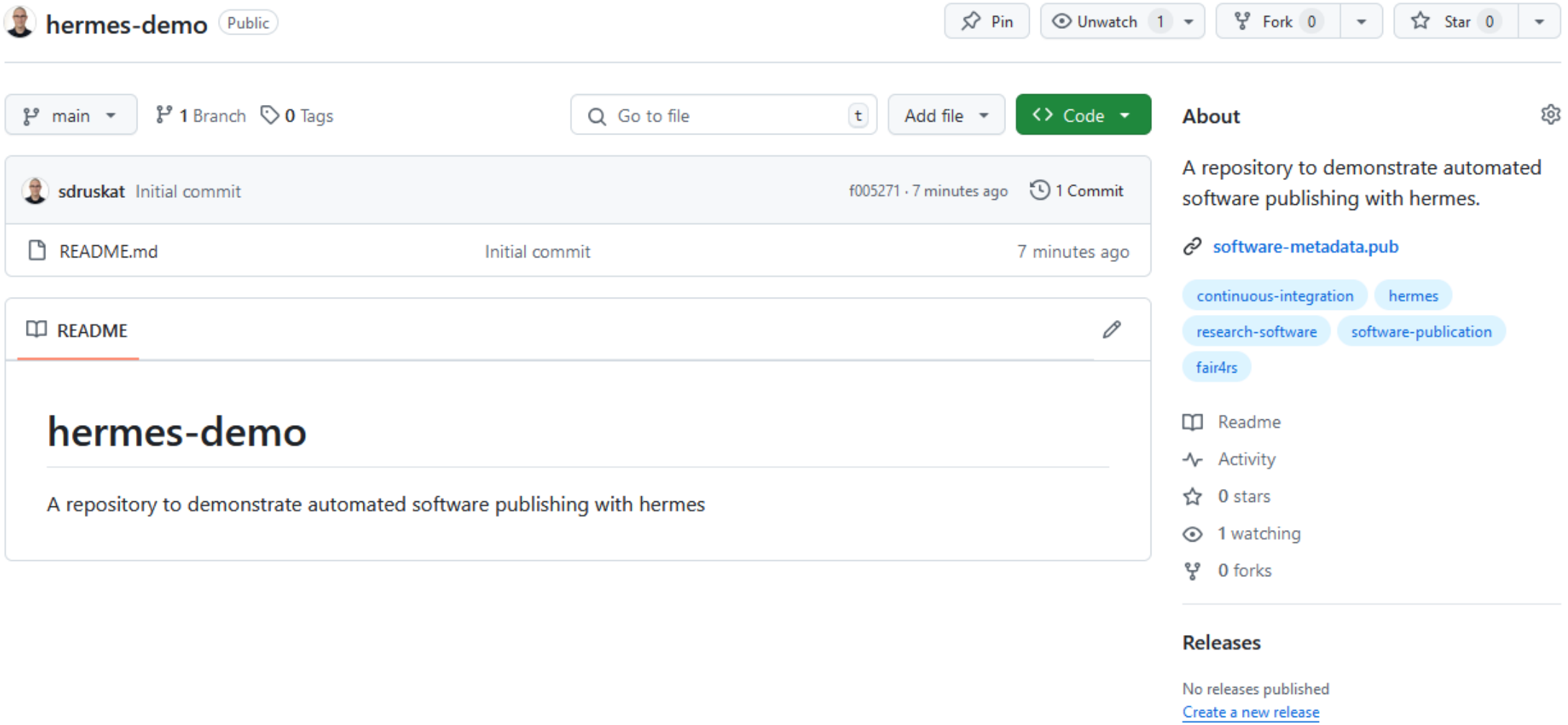
Meinel, M., Druskat, S., Kelling, J., Bertuch, O., Knodel, O., Pape, D., & Kernchen, S. (2024). hermes (v0.8.1). Zenodo. doi: 10.5281/zenodo.13311079.

Kernchen, S., Meinel, M., Druskat, S., Fritzsche, M., Pape, D., & Bertuch, O. (2024). Extending and Applying Automated HERMES Software Publication Workflows. arXiv. doi: 10.48550/arXiv.2410.17614.

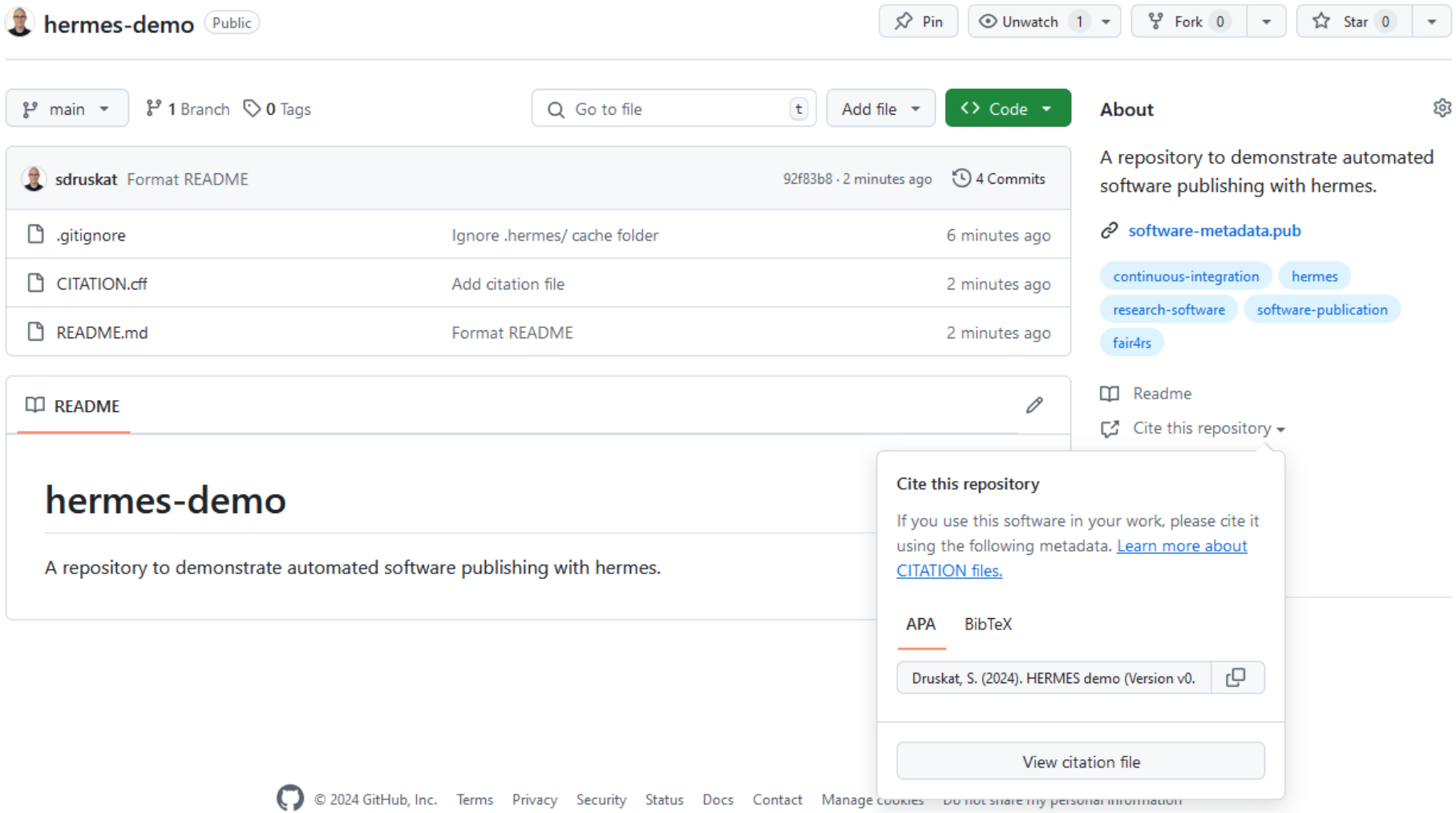
Fritzsche, M., & Meinel, M. (2024). hermes-plugin-python (0.2.0). Zenodo. doi: 10.5281/zenodo.13168127.

Jones et al. (2023). CodeMeta: an exchange schema for software metadata (Version 3.0). [Online]. Available: https://w3id.org/codemeta/v3.0

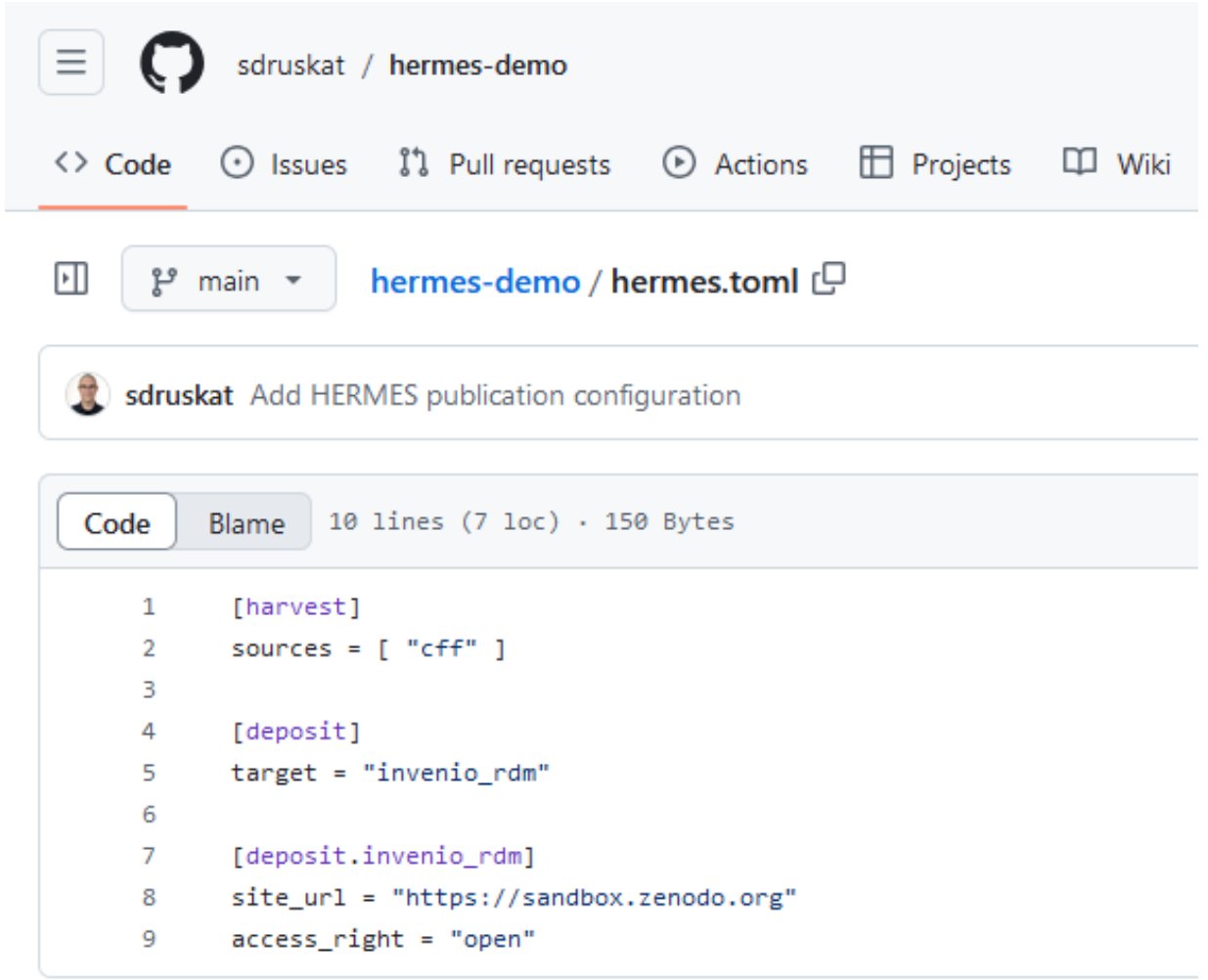
HERMES: Example
github.com/sdruskat/hermes-demo



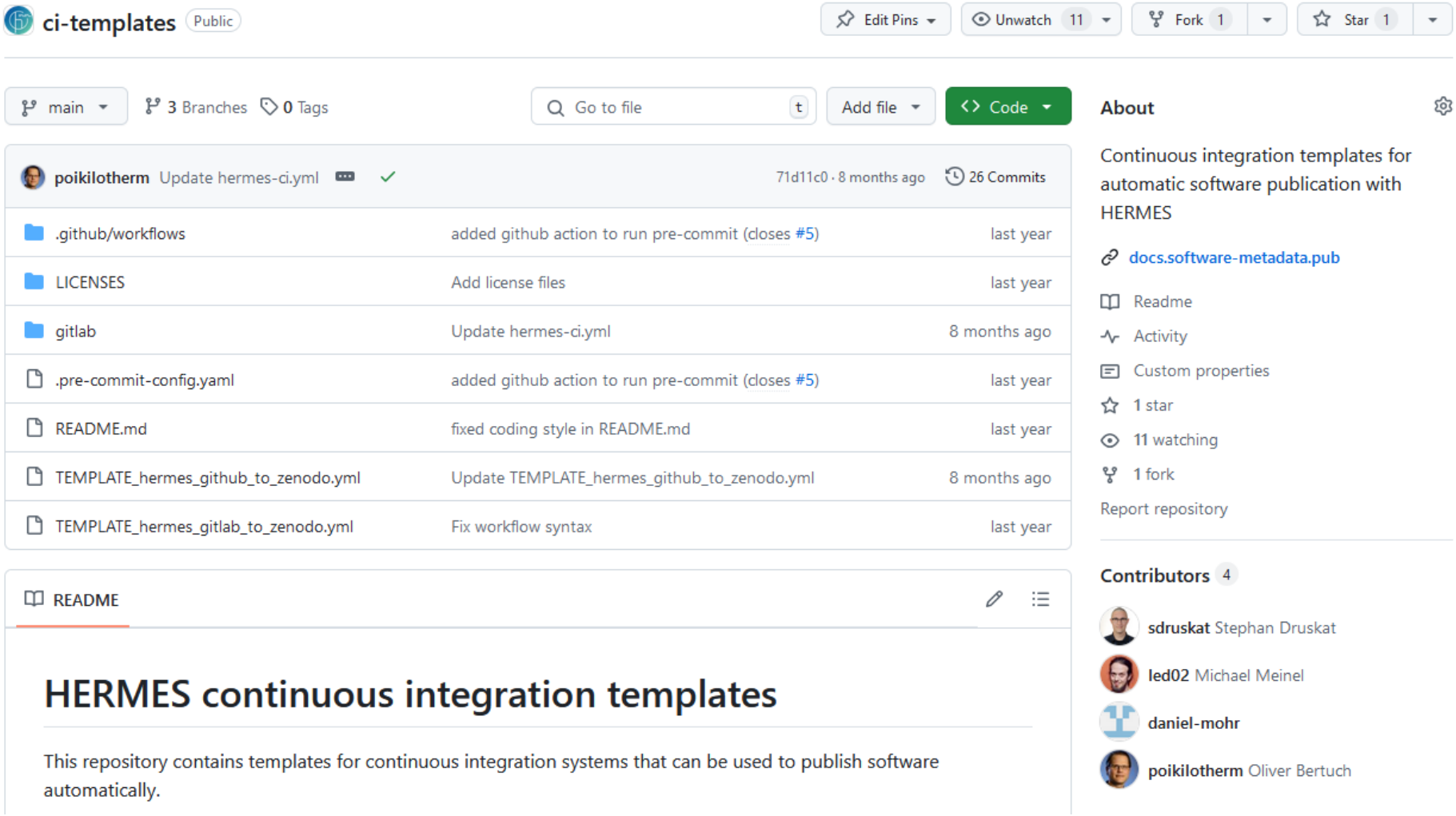
HERMES: Example
github.com/sdruskat/hermes-demo



HERMES: Example
github.com/sdruskat/hermes-demo



HERMES: Example
github.com/sdruskat/hermes-demo

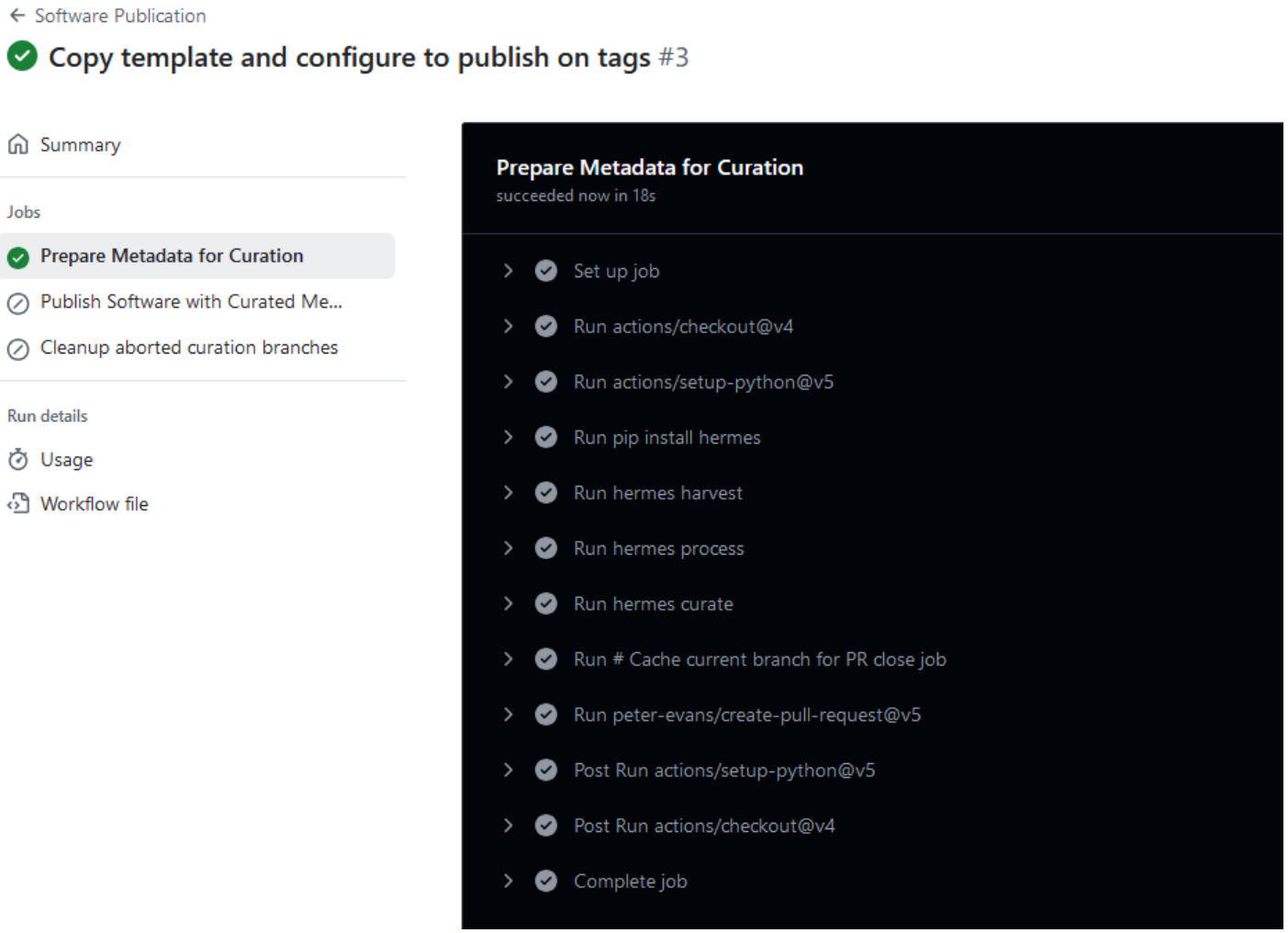


HERMES: Example
github.com/sdruskat/hermes-demo

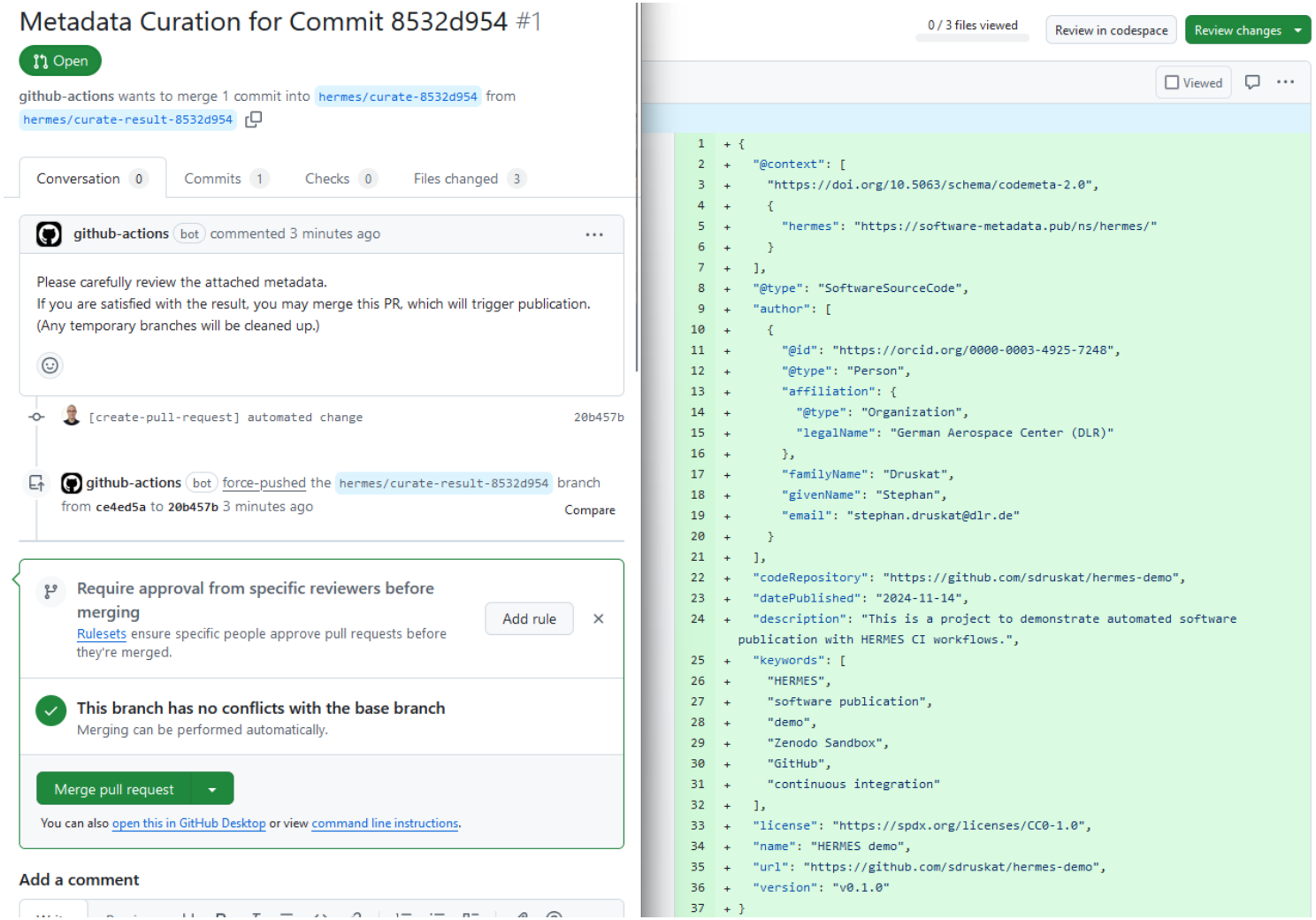
```
hermes-demo$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
hermes-demo$ git tag -a v0.1.0 -m "v0.1.0"
hermes-demo$ git push origin tag v0.1.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 162 bytes | 162.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:sdruskat/hermes-demo.git
 * [new tag]         v0.1.0 -> v0.1.0
hermes-demo$
```

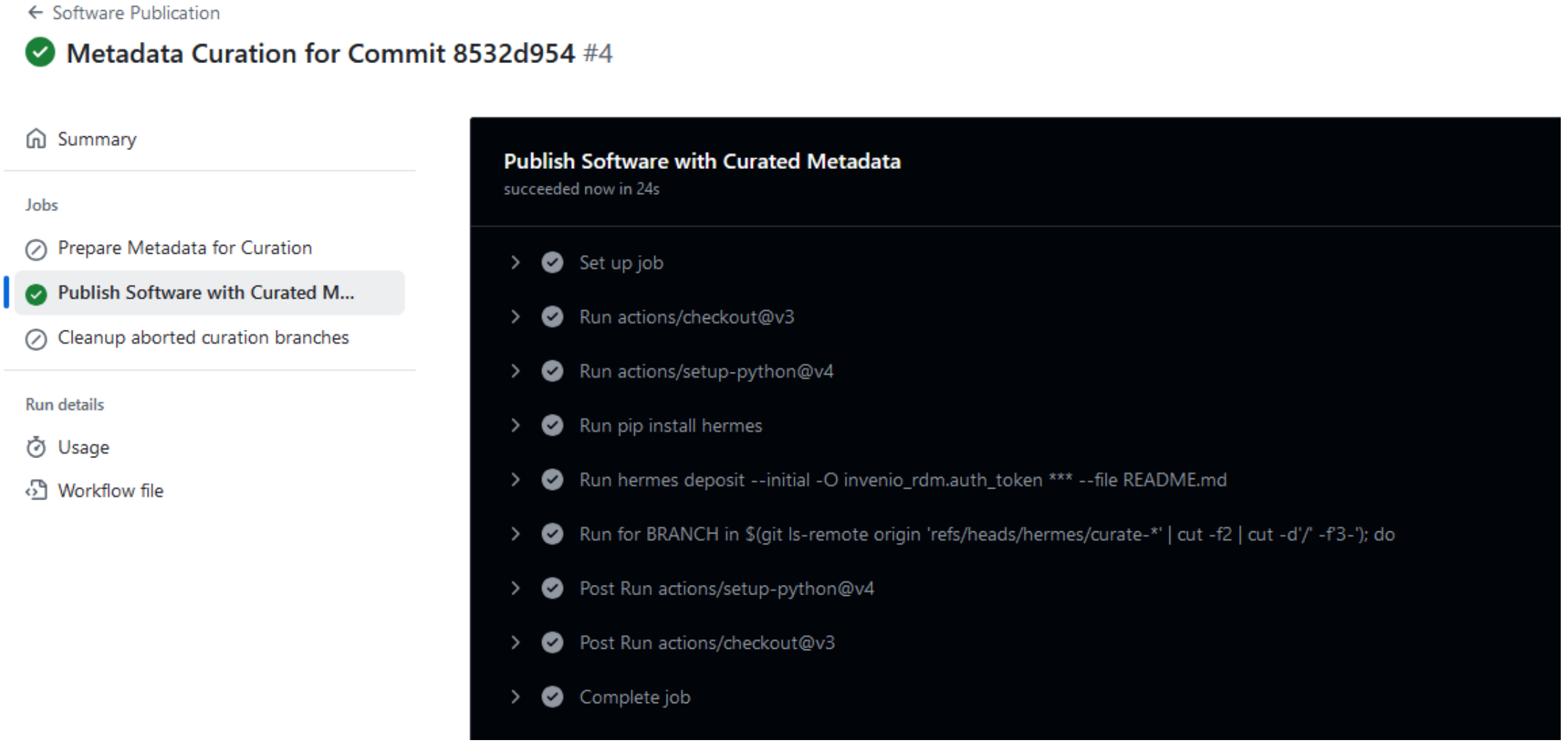
HERMES: Example
github.com/sdruskat/hermes-demo



HERMES: Example
github.com/sdruskat/hermes-demo



HERMES: Example
github.com/sdruskat/hermes-demo



HERMES: Example
github.com/sdruskat/hermes-demo