

Software Engineering 2022: rSE 22 – Research Software Engineering

## **HERMES: Automated software publication with rich metadata**

Stephan Druskat

*German Aerospace Center (DLR) & Humboldt-Universität zu Berlin*

with Oliver Bertuch (FZ Jülich), Oliver Knodel (HZDR), Guido Juckeland (HZDR), Tobias Schlauch (DLR)

# Overview

## of this presentation

---

- **Software publication**
  - **Why publish software?**
  - **What is the state of the art?**
- **HERMES**
  - **Where can HERMES help?**
  - **How does HERMES aim to help?**
  - **Software metadata**
  - **Target publication platforms**
  - **Project outputs**
- **Conclusion**

# Software publication

## The good news

---

- **Software is an important research output**
- **Ergo: RSEs do important work**
- **Publishing research software supports**
  - **Sustainability**
  - **Reproducibility**
  - **academic credit for RSEs**



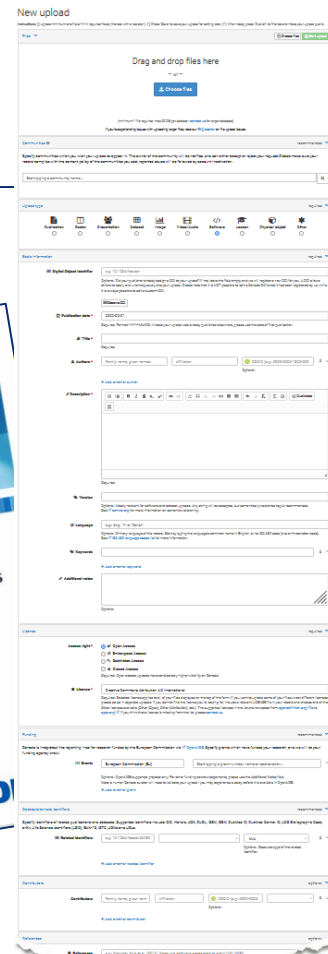
# Software publication

## The less good news

### To be FAIR:

1. It's a lot of work
2. It's a lot of manual work

- Identifiers (DOIs)
- Rich metadata
- Accessibility
- Machine-readable metadata
- Documenting dependencies
- Licenses
- Provenance
- ...
- Versions!



# Software publication

## What does it look like?

- **State-of-the-art satisfaction of FAIR principles:** metadata (+ software) in a publication repository
- **We have:** metadata, publication repositories

```
model = getattr(spectra, spectrum_dict["type"])

if norm.unit in (u.Unit("erg"), u.Unit("erg cm-3")) and norm_type != "integral":
    raise NameError(
        "Normalisation different than 'in'"
    )

# check the units of the normalisation
# cm-3 is the only one allowing more than
if norm.unit == u.Unit("cm-3"):
    if norm_type == "differential":
        final_model = model(norm, **spect
    elif norm_type == "gamma-1":
        final_model = model.from_norm_at_
        norm, **spectrum_dict["parame
    )
    elif norm_type == "integral":
        final_model = model.from_normalis
        norm, **spectrum_dict["parame
    )

{
  "@context": "https://doi.org/10.5063/schema/codemeta-2.0",
  "@type": "SoftwareSourceCode",
  "license": "https://spdx.org/licenses/BSD-3-Clause",
  "codeRepository": "https://github.com/cosimonigro/agnpy",
  "contIntegration": "https://github.com/cosimonigro/agnpy/actions",
  "dateCreated": "2019-12-17",
  "datePublished": "2022-01-31",
  "dateModified": "2021-08-02",
  "downloadUrl": "https://github.com/cosimonigro/agnpy/releases/tag/v0.1.6",
  "issueTracker": "https://github.com/cosimonigro/agnpy/issues",
  "name": "agnpy",
  "version": "0.1.8",
  "identifier": "10.5281/zenodo.4955175",
  "description": "agnpy is a python package focusing on the computation of the",
  "applicationCategory": "astrophysics",
  "funding": "ESCAPE EU H2020 824064",
  "developmentStatus": "active",
  "isPartOf": "https://www.astropy.org/affiliated/#affiliated-packages",
```



January 31, 2022

agnpy

Nigro, Cosimo; Starek, Julian; Gherty, Pawel; Sanchez, David; Drag, Matheo; Vulliamis, Thomas

agnpy is a python package focusing on the computation of the radiative processes of relativistic particles accelerated in the jets of Active Galactic Nuclei (AGN). It includes classes describing the galaxy components responsible for line and thermal emission and calculates the absorption due to gamma-gamma pair production on soft (IR-UV) photon fields.

Preview

agnpy-v0.1.6.zip

cosimonigro-agnpy-d4bd022

github

- workflows
- upload.yml
- test.yml

gtpnone

pykic

zenodo.json

LICENSE

MANIFEST.in

README.md

agnpy

- \_\_init\_\_.py
- absorption
- \_\_main\_\_.py
- absorption.py
- compton

705 Bytes

1.2 kB

403 Bytes

18.5 kB

1.9 kB

1.5 kB

322 Bytes

2.4 kB

202 Bytes

26 Bytes

30.3 kB

Files (5.3 MB)

Name	Size	Preview	Download
cosimonigro/agnpy-v0.1.6.zip	5.3 MB		
md51b1e0c224335a6a30b10b1eaf0b2c4			

3 Citations

Show only: Literature (3) Unknown (1) Dataset (0) Software (0)

Search

VHE gamma-ray detection of PSR B202+326 and modeling ...  
Auer, V. A. et al. (DOI: 10.1051/0004-6361/202039687)

2021

agnpy: an open-source python package modelling the radiat...  
Nigro, C. et al.

2021

PyRemodel: An open-source Python package for one-zone num...  
Dallier, Y. et al.

2021

ADS: 2021ArXiv121112928M

768 views

157 downloads

See more details...

Available in

GitHub

OpenAIRE

Publication date: January 31, 2022

DOI: [10.5281/zenodo.4955175](https://doi.org/10.5281/zenodo.4955175)

Keywords:

ESCAPE

ESCAPE - European Science Cluster of Astronomy & Particle physics (ESAP)

research infrastructures (824064)

Related identifiers:

Supplement to <https://github.com/cosimonigro/agnpy/tree/v0.1.6>

Communities:

ESCAPE 2020

License (for files):

CC-BY 3.0 - Attribution 3.0 - Revised License

Versions

Version	Jan 31, 2022
Version 0.1.8	10.5281/zenodo.4955175
Version 0.1.7	10.5281/zenodo.3927797

# Software publication

## Where can HERMES help?

---

- **We have:**

- Publication repositories with metadata input forms/pull-based workflows
- Metadata from different sources, in different formats/modes

- **We want:**

- As little to do with forms as possible
- Control over what goes into the publication (push-based workflows)
- A collated set of rich metadata for publication

## Automated software publication with rich metadata

- 
- HELMHOLTZ
- 
- PROJECT HERMES

# Software publication with HERMES

## Scope

The user receives assistance in depositing software in an automated fashion. This may be used to create publications purely with rich metadata (to be at least FAIR [5], even for closed source software) or with attached artifacts like source code, executables, etc. (to be more easily reusable). To achieve this, HERMES provides

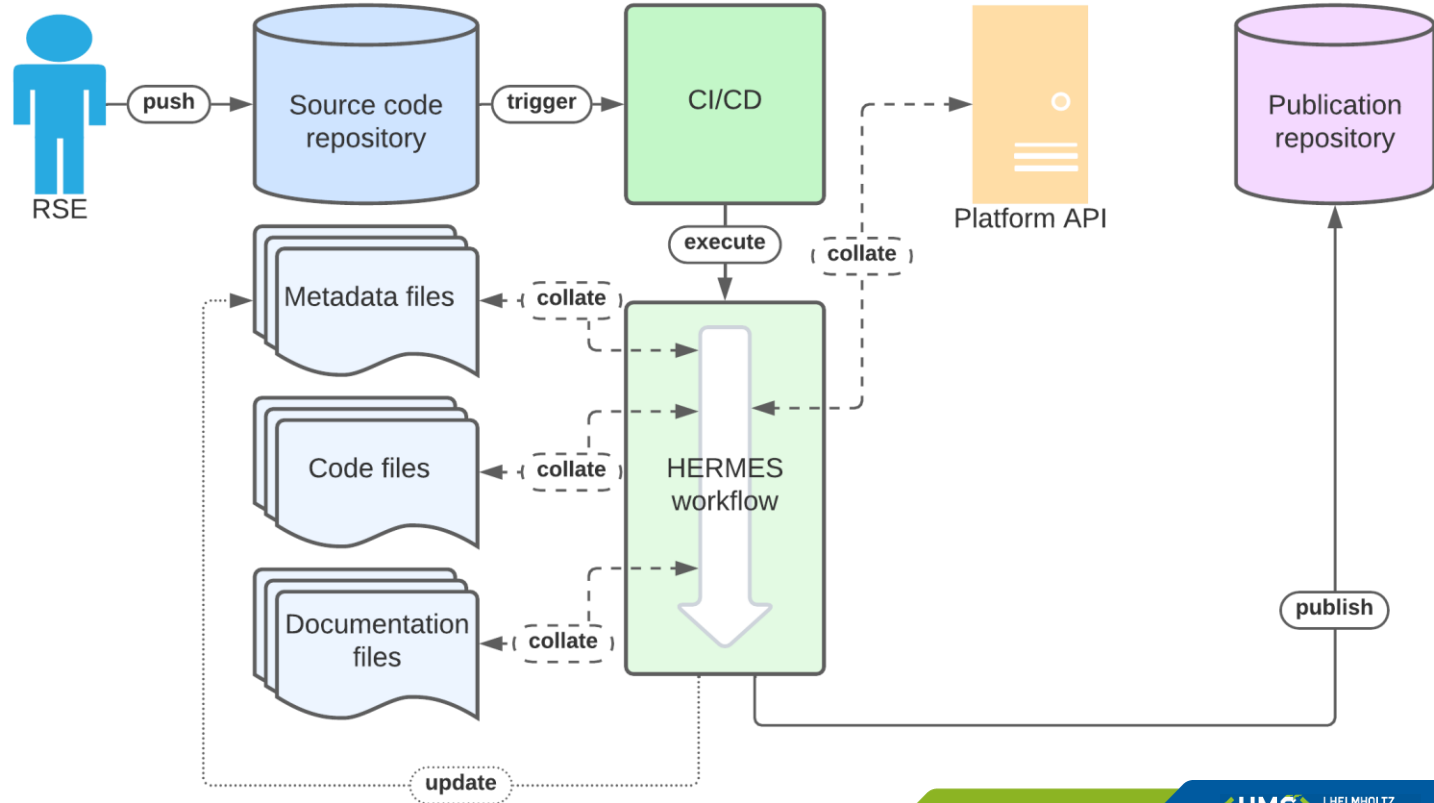
- an extensible, configurable and automatable toolchain with capability to be executed for<sup>15</sup>
  - N software publications in
  - M target publication repositories
  - from the same origin
  - as configured by the user,
- initially harvesting and collating **statically available metadata** from formerly described **metadata sources** and
- initially targeting
  - **InvenioRDM** and
  - **Dataverse project**
- for deposits of metadata and artifacts according to curator-defined requirements
- and output of the respective metadata in a structured format (e.g., **CodeMeta files**) for further reuse.

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, [abs/2201.09015](https://arxiv.org/abs/2201.09015).



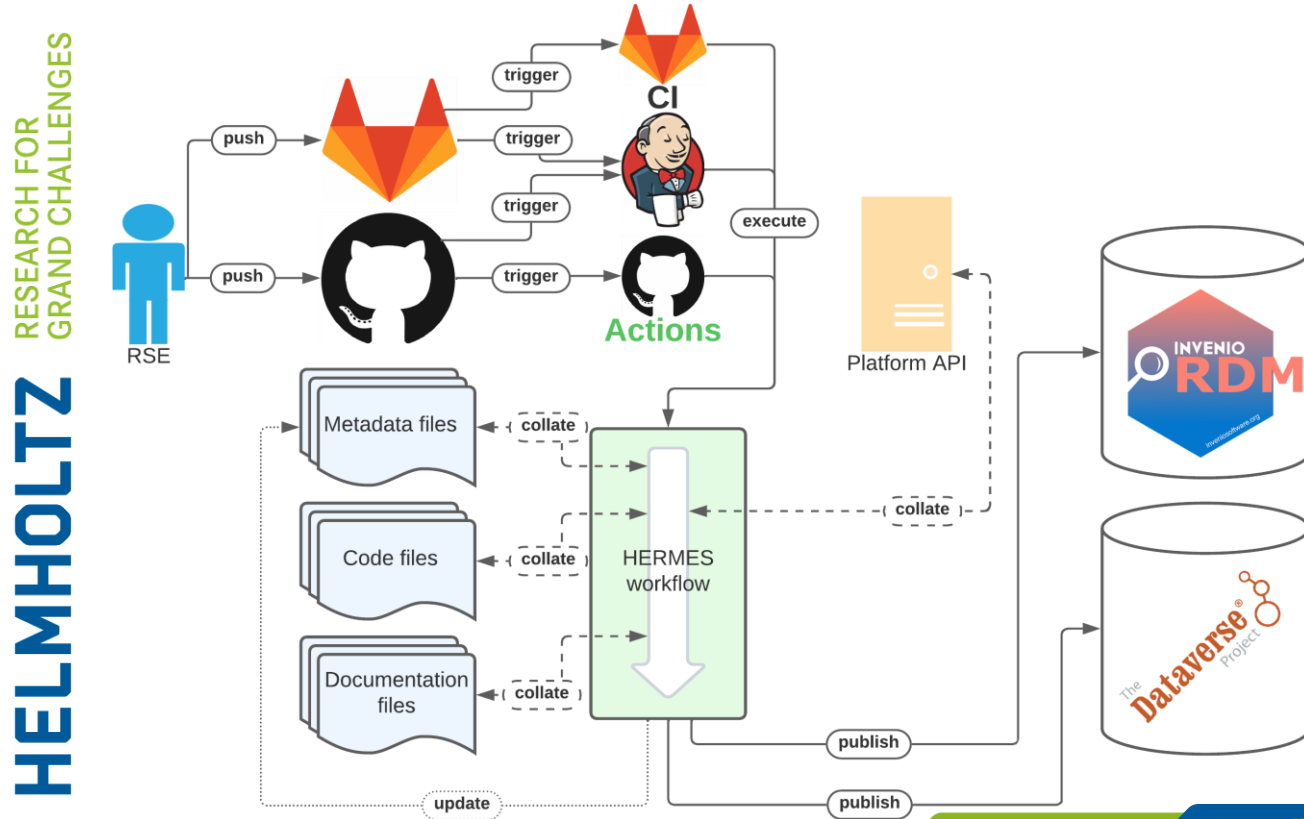
# Software publication with HERMES

## Overview (simplified)



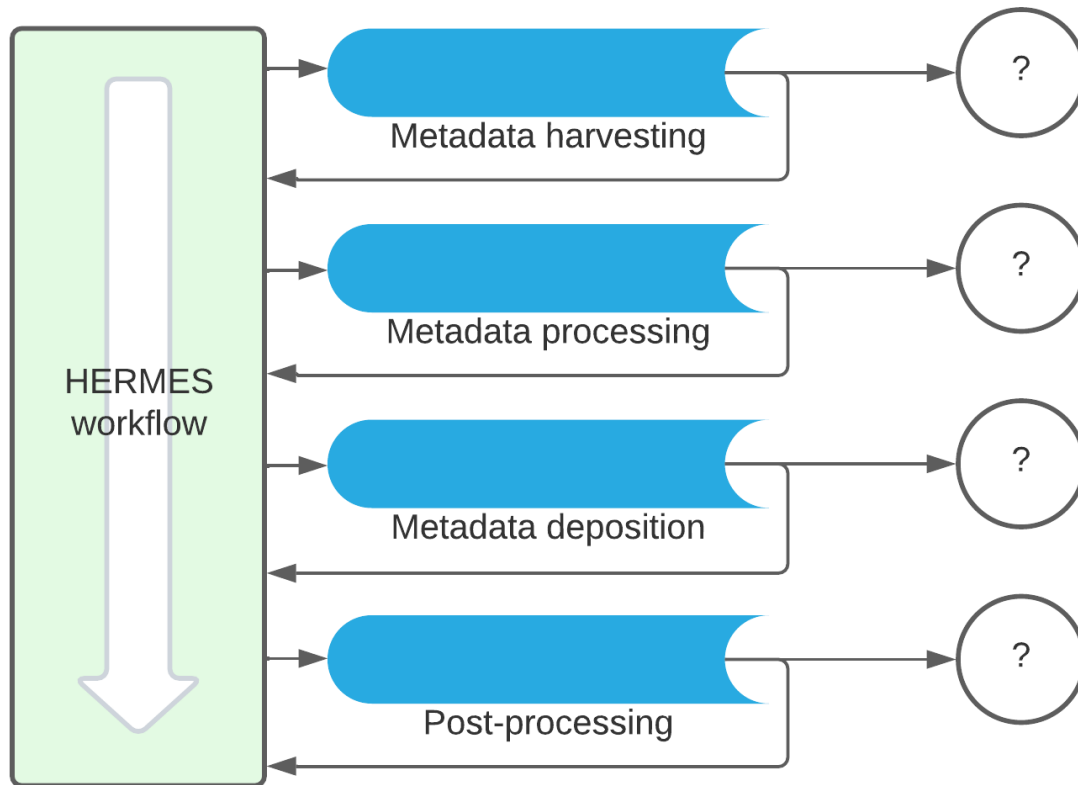
# Software publication with HERMES

## Overview (simplified and concretized for HERMES Mk. I)



# Software publication with HERMES

## Workflow pipelines



# Software publication with HERMES

## Metadata: types, formats, sources for HERMES

---

- **Metadata**

- Differences in generation, scope, mode, aspects
- Generic software metadata vs. software-specific metadata

- **Metadata formats**

- Metadata files, snippets, third-party systems, API responses
- Structured vs. unstructured

- **Sources**

1. Collectable structured metadata
2. (Metadata from minable structured data)
3. (Metadata from minable unstructured data)

# Software publication with HERMES

## Depositing on target publication platforms

---

- **Preparation of repository software**
  - Prepare Dataverse project & InvenioRDM
- **Curator-defined requirements**
  - Query requirements for deposition
  - Feedback loop



# What will HERMES give you?

## Project outputs

---

- **Software**
  - Software for software publication workflow automation
- **CI templates**
  - GitLab CI, GitHub Action, Jenkins, Travis CI
- **Training materials**
  - Adaption of HIFIS training materials to include workflow usage
- **Project website**
  - One-stop shop for information and documentation
- **Policy proposals**
  - Proposals for updates to policies/ guidelines at Helmholtz and Allianzinitiative

# Conclusion: what problem does HERMES solve?

Advancing the state of the art in software publication

---

- **Enable automated publication of research software according to the FAIR principles** (even for non-open source software)
- Automate collation of metadata from different sources
- Automated synchronization of software metadata between publication and repository
- Differentiated treatment of repositories containing >1 package
- Differentiated treatment of software, documentation and data components

# Thank you!

## Where to learn more about project HERMES?

---



Stephan Druskat, DLR, PI, [@stdruskat](#)



Oliver Bertuch, FZJ, PI, [@poi\\_ki\\_lo\\_therm](#)



Guido Juckeland, HZDR, PI, [@GuidoJuckeland](#)



Oliver Knodel, HZDR, [@olikno1](#)



Tobias Schlauch, DLR, [@TobiasSchlauch](#)

- Find us on [Twitter](#)
- Write an email to [team@software-metadata.pub](mailto:team@software-metadata.pub)
- Go to [software-metadata.pub](http://software-metadata.pub)

# Software publication with HERMES

## Dealing with different source code repository layouts

---

### 1. Ideal state

- Repository contains 1 software package with integrated software documentation

### 2. god repositories

- Repository contains  $>1$  software package with integrated software documentation

### 3. Mixed repositories

- Repository contains  $\geq 1$  software package with integrated software documentation,  $\geq 1$  dataset,  $\geq 1$  manuscript

**HERMES lets users specify which repository parts to include for publication**

# Software publication with HERMES

## Metadata: tooling

---

### ■ Tooling

- Software Metadata Extraction Framework ([SoMEF](#))
- CaltechDATA Automated Metadata Service ([AMES](#))
- [codemeta2cff GitHub Action](#)
- [CodeMeta Crosswalks](#)
- Citation File Format [Converter](#) and [GitHub Action](#)
  - Other [CFF tooling](#)
- Citation File Format [Initializer](#)



# Software publication with HERMES

## Workflows: potential building blocks

---

### ■ Tooling

- Depositioning on Zenodo ([zenodraft](#) and [GitHub Action](#))
- Software Heritage Save [GitHub Action](#)
- Software Heritage Deposit [Command Line Tool](#)
- Dataverse Uploader [GitHub Action](#)