Software Heritage

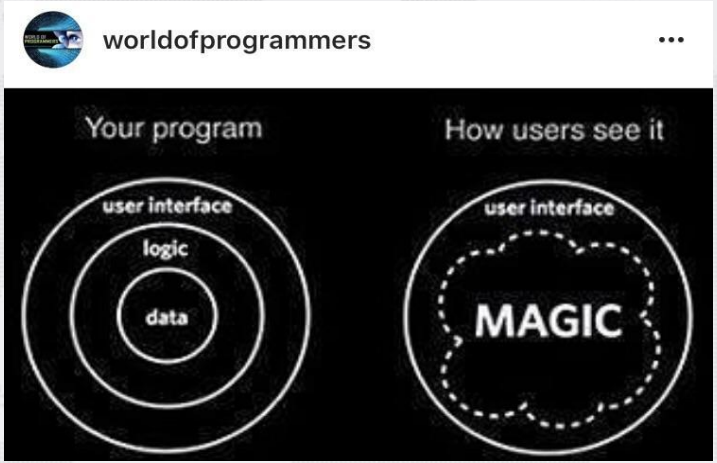
THE GREAT LIBRARY OF SOURCE CODE

# Challenges and opportunities in Academia towards Open Source software

# Enjeux de l'ouverture des logiciels académiques

Morane Gruenpeter
Director of Scholarly Ecosystems
Software Heritage, Inria Foundation

# What is software?




Software as a concept

- **project** or entity
- the **community** around the project
- the software **idea** / algorithms / solutions

Software artifacts

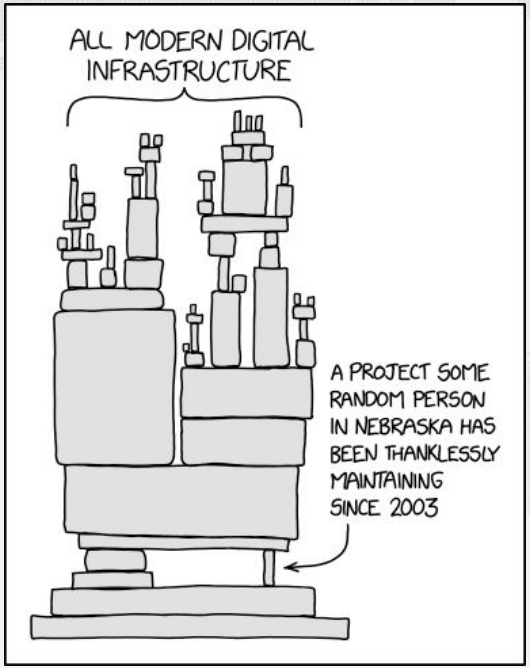
- Executables
  - For multiple environments
- Source code

*Not a digital artifact*

*A very large collection of digital artifacts*

# Source code is…

## The invisible backbone of all digital infrastructure




https://xkcd.com/2347/

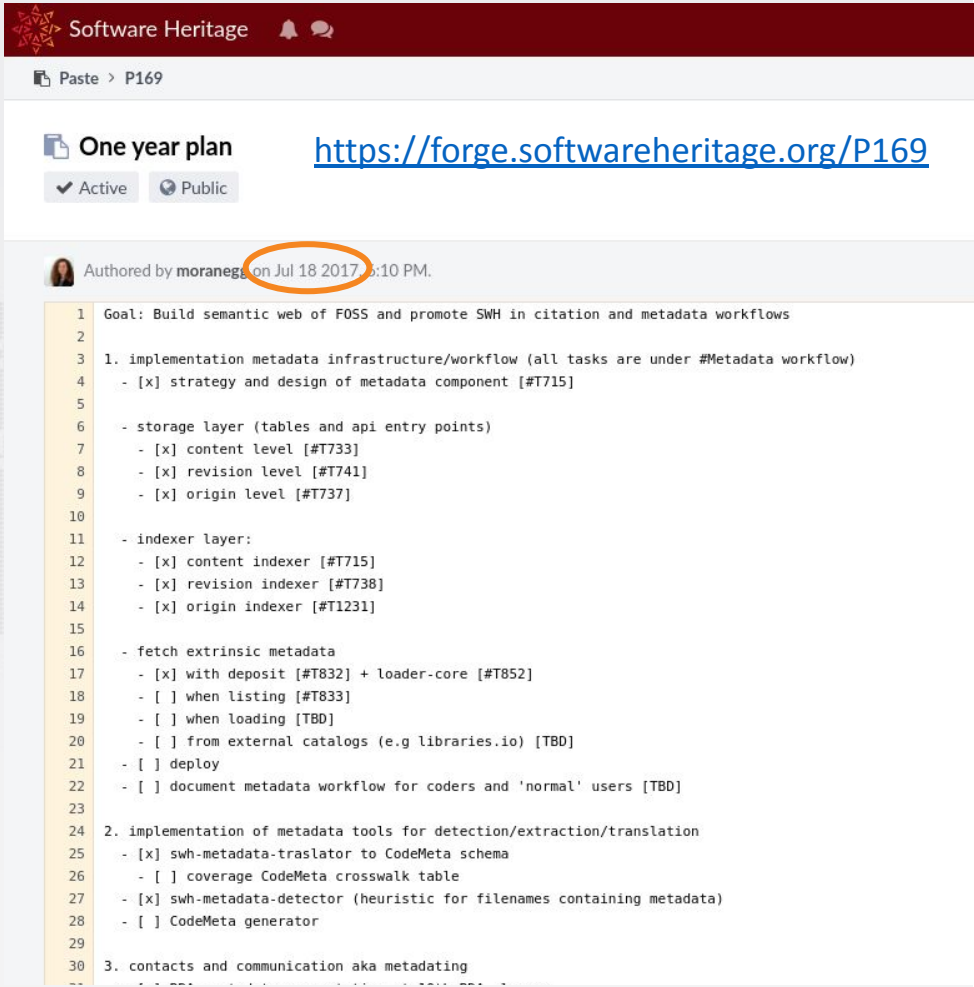
Is #DigitalSovereignty possible without it?

# Goals

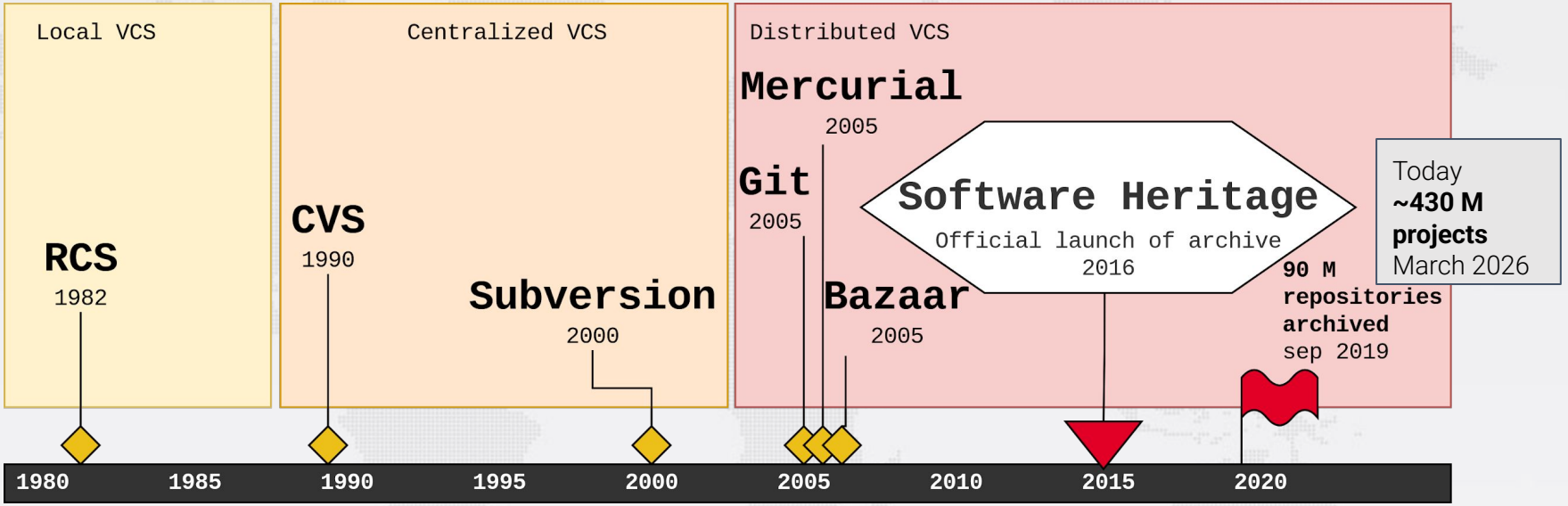
- ★ **Introduction: why are we here?**
- ★ Open Science isn’t complete without Open Source
- ★ Guided exploration: into the Software Heritage archive
- ★ Good practices for #BetterMetadata
- ★ The role of libraries: collaborating with OSPOs

# Why am I here?

1. **Raise awareness** about the universal source code archive.
2. **Share** vision on **Openness** in research and in software engineering
3. **Promote software citation** and metadata workflows
4. **Build bridges** between communities, especially with **Academia.**



# Version control in the software forges era




- records changes made to a (set of) source code file (s)
- allows to operate on versions: diff/merge/fork/recover etc.
- essential tool for software development

# Free/Libre and Open Source Software

## "Open" versus "free"

*Free and open-source software (FOSS) or free/libre and open-source software (FLOSS) is openly shared source code that is licensed without any restrictions on usage, modification, or distribution.[citation needed] Confusion persists about this definition because the **"free"**, also known as **"libre"**, refers to the **freedom of the product**, not the price, expense, cost, or charge. For example, "being free to speak" is not the same as "free beer".[19]*

https://en.wikipedia.org/wiki/Open_source

https://www.gnu.org/philosophy/free-sw.html


https://openclipart.org/detail/246027/four-freedoms-of-free-software




https://openclipart.org/detail/320626/craft-beer-glass

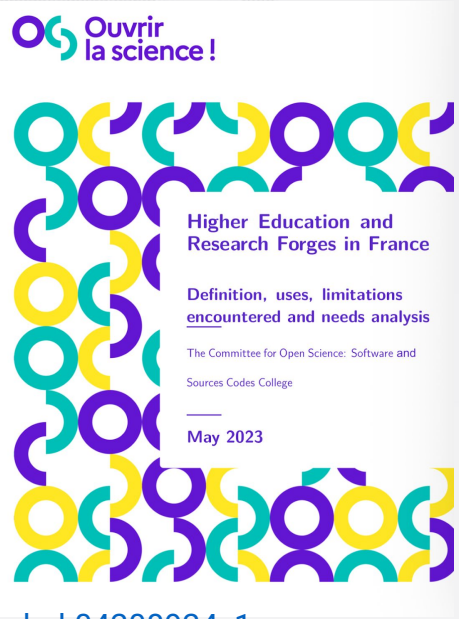
# Licensing is a major part in Open Source


https://www.tcpipacademy.com/open-source-licensing/
https://www.computernetworkingnotes.com/networking-tutorials/software-licenses-types-explained.html

# What are software forges?

*"**Software forges** are mainly designed to manage software-engineering artefacts (such as the source code), the binary files and even the documentation over the entire life cycle. Forges can also be used to collaboratively draft scientific papers or documentation, or even write web pages with the help of tools for processing poorly structured text files …"*

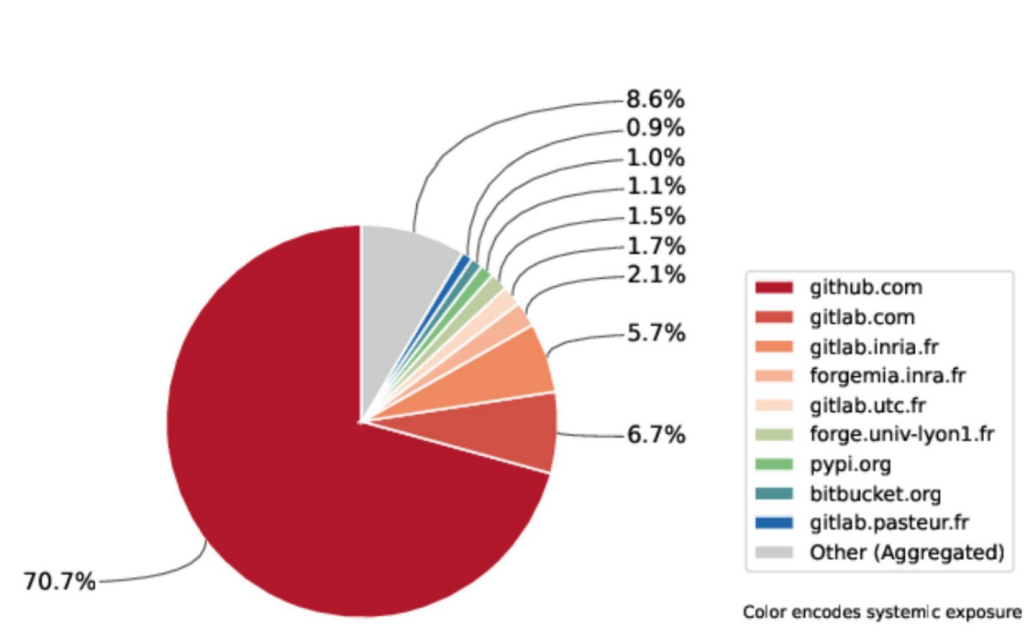
GitHub




hal-04208924v1

# Tracking researchers commits

## Infrastructure concentration (French academia)




Di Cosmo, 2025, French Academia Insights report

Massive concentration in

**GitHub > 70%**

# The commons definitions

**The commons**

*The commons are the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable Earth. These resources are held in common even when owned privately or publicly.*

https://en.wikipedia.org/wiki/Commons (retrieved March 1st.)

**The software commons**

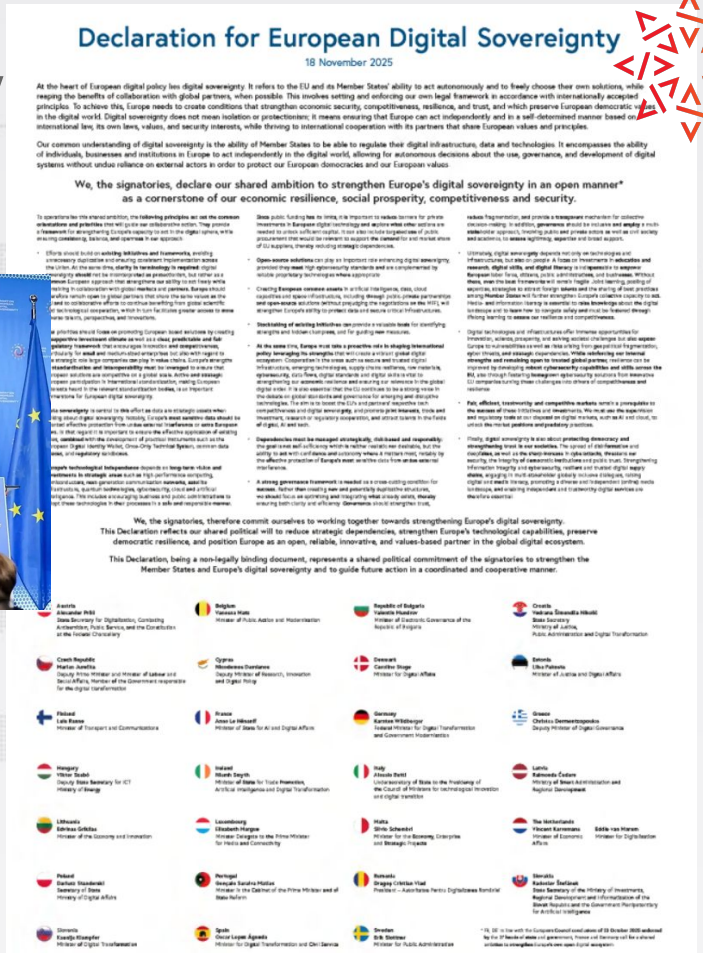
*The software commons[1] consists of all computer software which is available to everyone to use for any purpose. It includes licensed open source software which can be used, studied, modified, and redistributed with few, if any, obligations.[2] It also includes software in the public domain that is not technically open source software.*

https://en.wikipedia.org/wiki/Software_Commons (retrieved March 1st.)

# Defining Digital Sovereignty

## Key aspects from the Berlin declaration


- ★ Strategic Autonomy
  - Reduced Dependencies
- ★ Digital Commons
  - Open Source Adoption
- ★ Investment in Strategic Technologies
  - AI, Cloud, Cybersecurity



Declaration for European Digital Sovereignty

18 November 2025

At the heart of European digital policy lies digital sovereignty. It refers to the EU and its Member States' ability to act autonomously and to freely choose their own solutions, while reaping the benefits of collaboration with global partners, when possible. This involves setting and enforcing our own legal framework in accordance with internationally accepted principles. To achieve this, Europe needs to create conditions that strengthen economic security, competitiveness, resilience, and trust, and which preserve European democratic values in the digital world. Digital sovereignty does not mean isolation or protectionism; it means ensuring that Europe can act independently and in a self-determined manner based on our international law, its own laws, values, and security interests, while thriving to international cooperation with its partners that share European values and principles.

Our common understanding of digital sovereignty is the ability of Member States to be able to regulate their digital infrastructure, data and technologies. It encompasses the ability of individuals, businesses and institutions in Europe to act independently in the digital world, allowing for autonomous decisions about the use, governance, and development of digital systems without undue reliance on external actors in order to protect our European democracies and our European values.

We, the signatories, declare our shared ambition to strengthen Europe's digital sovereignty in an open manner* as a cornerstone of our economic resilience, social prosperity, competitiveness and security.

We, the signatories, therefore commit ourselves to working together towards strengthening Europe's digital sovereignty.
This Declaration reflects our shared political will to reduce strategic dependencies, strengthen Europe's technological capabilities, preserve democratic resilience, and position Europe as an open, reliable, innovative, and values-based partner in the global digital ecosystem.

This Declaration, being a non-legally binding document, represents a shared political commitment of the signatories to strengthen the Member States and Europe's digital sovereignty and to guide future action in a coordinated and cooperative manner.

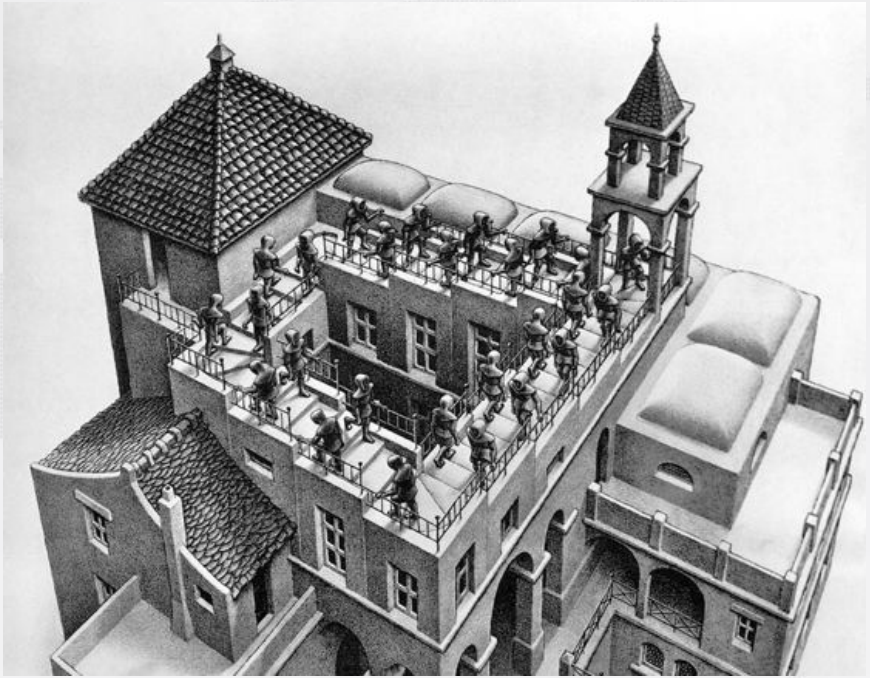
Source – Federal Government of Germany

# EUtopia or Dystopia for Software

Is #DigitalSovereignty even possible?

Is it aligned with the Open Science and Open Source vision?

From Policies to Infrastructures: the strange loop is infinite.



*Ascending and Descending* by M. C. Escher

# Qu’est-ce qui vous amène aujourd’hui ?

Sur menti:  6268 6204

# Le logiciel est-il inclus dans la stratégie Science Ouverte de votre institution ?

Posibilités

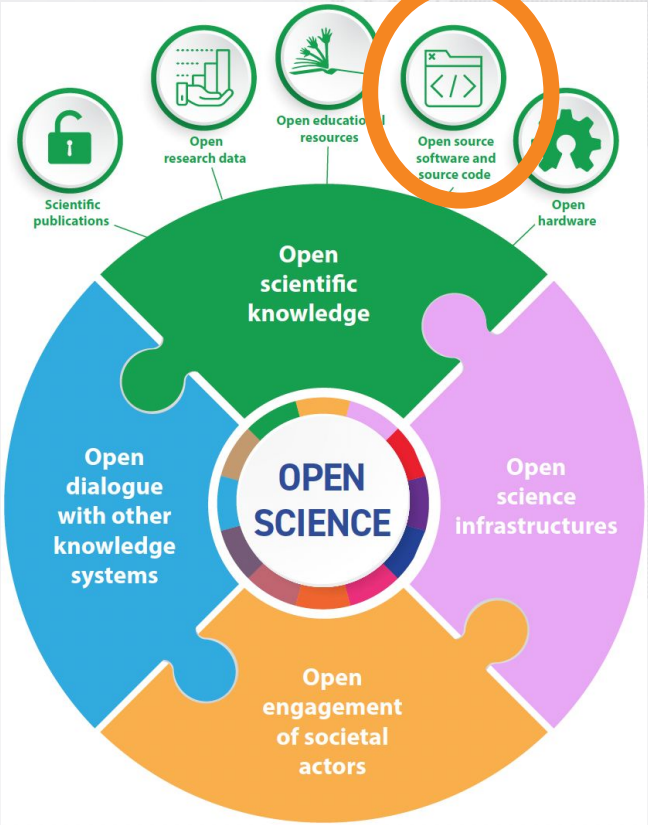
1. Forte inclusion
2. Faible inclusion
3. Pas de logiciel dans la politique Science Ouverte
4. Pas de politique Science Ouverte

Sur menti: 6268 6204

# Goals

- ★ Introduction: why are we here?
- ★ **Open Science isn't complete without Open Source**
- ★ Guided exploration: into the Software Heritage archive
- ★ Good practices for #BetterMetadata
- ★ The role of libraries: collaborating with OSPOs

# A vision: Open Science for knowledge freedom




"Building on the essential principles of academic freedom, research integrity and scientific excellence, open science sets a new paradigm that integrates into the scientific enterprise practices for **reproducibility**, **transparency**, **sharing** and **collaboration** resulting from the increased opening of scientific contents, tools and processes."

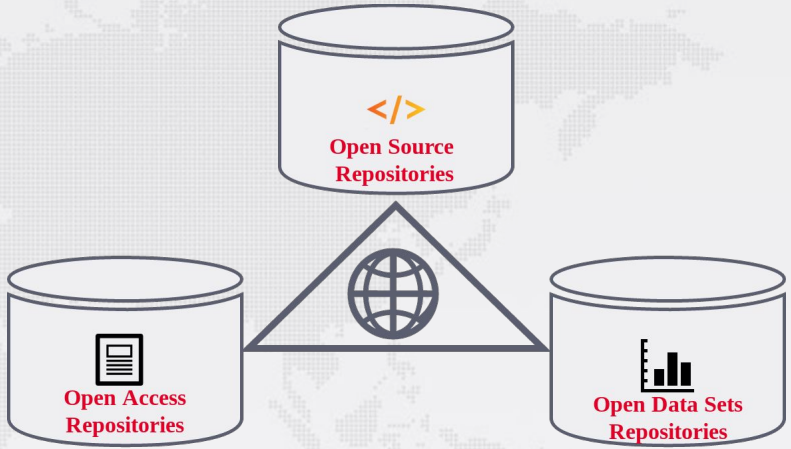
UNESCO Recommendations

# Software in Research: A pillar of Open Science

Multiple facets, it can be seen as:

- a tool
- a research outcome or result
- the object of research

Tim Berners-Lee the World Wide Web was born in Academia, 1989,at CERN




*Three pillars of Open Science*
*Gruenpeter, Software Heritage CC-By 4.0 2019*

# The Knowledge is in the Source Code

*“Source code provides a view into the mind of the designer.”*

Len Shustek, 2006
Computer History Museum

Apollo 11 Guidance Computer (~60.000 lines), 1969

```
P63SPOT3        CA      BIT6            # IS THE LR ANTENNA IN POSITION 1 YET
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     P63SPOT4        # BRANCH IF ANTENNA ALREADY IN POSITION 1

                CAF     CODE500         # ASTRONAUT:    PLEASE CRANK THE
                TC      BANKCALL        #               SILLY THING AROUND
                CADR    GOPERF1
                TCF     GOTOPOOH        # TERMINATE
                TCF     P63SPOT3        # PROCEED       SEE IF HE'S LYING

P63SPOT4        TC      BANKCALL        # ENTER         INITIALIZE LANDING RADAR
                CADR    SETPOS1

                TC      POSTJUMP        # OFF TO SEE THE WIZARD ...
                CADR    BURNBABY
```

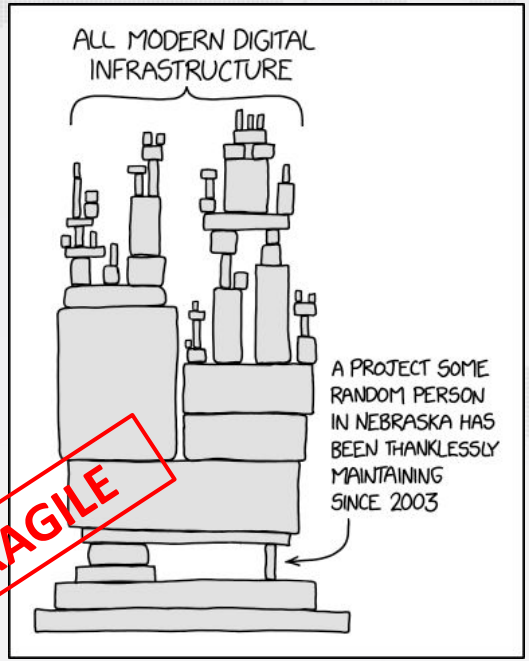
Margaret Hamilton

Apollo 11 source code in SWH | Article

Preserved in the **MIT research library**
digitized by Rob Burkey since 2003

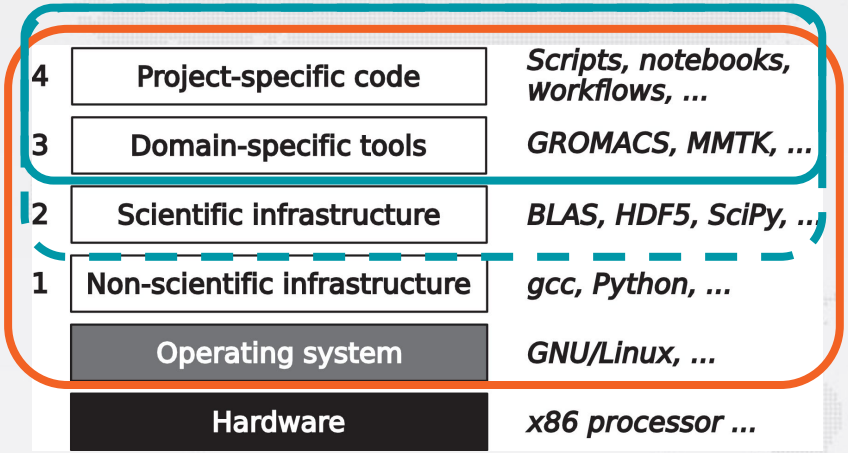
# The software stack: More than Research Software

Research Software is a thin layer on top of the global software stack




https://xkcd.com/2347/

## What's the the Software Collapse?

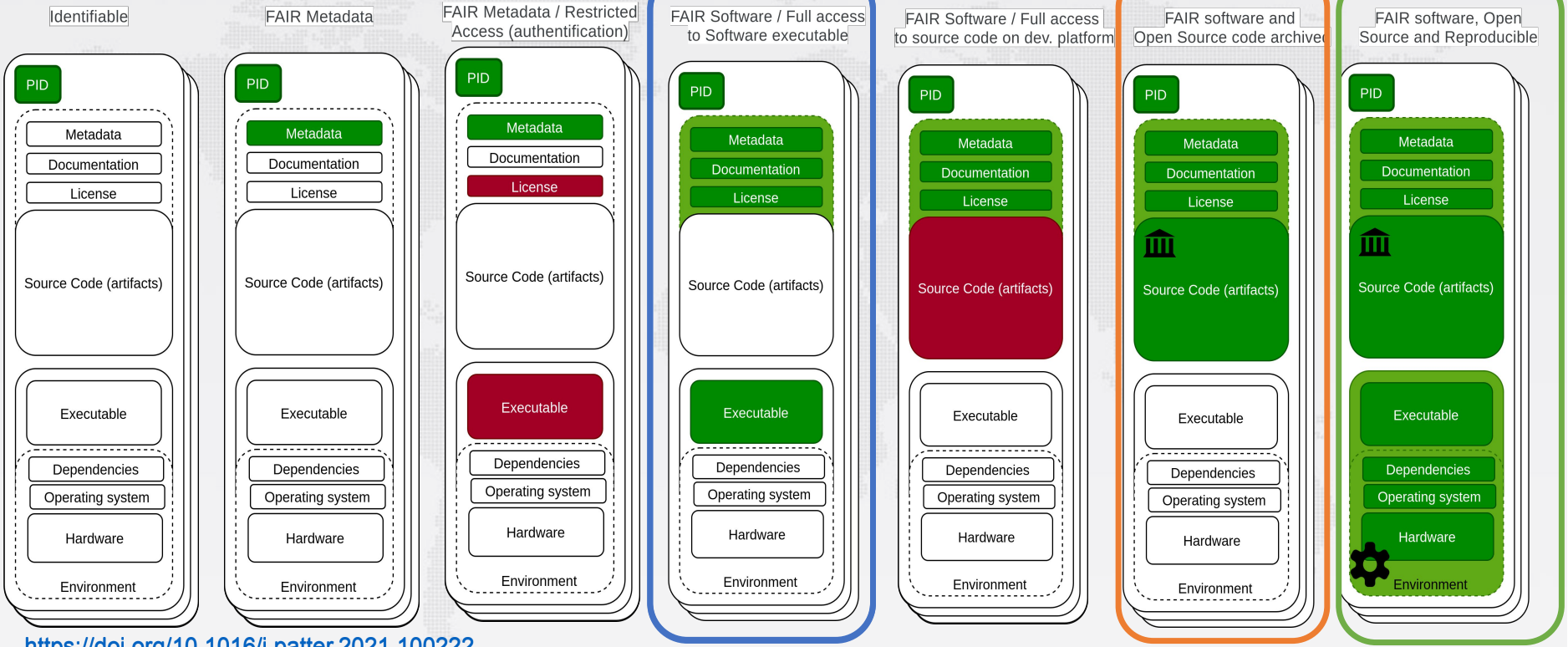



Konrad Hinsen. Dealing With Software Collapse. Computing in Science and Engineering, 2019, 21 (3), pp.104-108. ⟨10.1109/MCSE.2019.2900945⟩. ⟨hal-02117588⟩

Can be preserved in a:

- Scholarly Infrastructure
- Universal Source Code Archive

# FAIR, Open Source and Open Science?




https://doi.org/10.1016/j.patter.2021.100222

# Why Open Source is essential to Open Science

**Reproducibility**: Without open code, results cannot be fully verified or replicated.

**Transparency**: Algorithms and workflows must be inspectable.

**Reusability**: Open licenses allow others to reuse, adapt, and extend research tools.

**Collaboration**: Open development enables global, distributed contribution. Building on communities and governance.

**Digital Sovereignty**: Institutions choose how to retain control over their research assets

**Sources**
The Good Governance Handbook https://ospo-alliance.org/ggi/
A tale of two 'opens': intersections between Free and Open Source Software and Open Scholarship. SocArXiv. doi:10.31235/osf.io/2kxq8

# Defining Research Software: Still a controversy?

Research Software includes source code files, algorithms, scripts, computational workflows and executables that were **created during the research process or for a research purpose**. Software components (e.g., operating systems, libraries, dependencies, packages, scripts, etc.) that are used for research but were not created during or with a clear research intent should be considered software in research and not Research Software. This differentiation may vary between disciplines. The minimal requirement for achieving computational reproducibility is that all the computational components (Research Software, software used in research, documentation and hardware) used during the research are identified, described, and made accessible to the extent that is possible.

**Research Software ≠ Software in research**

Gruenpeter, M., Katz, D. S., Lamprecht, A.-L., Honeyman, T., Garijo, D., Struck, A., Niehues, A., Martinez, P. A., Castro, L. J., Rabemanantsoa, T., Chue Hong, N. P., Martinez-Ortiz, C., Sesink, L., Liffers, M., Fouilloux, A. C., Erdmann, C., Peroni, S., Martinez Lavanchy, P., Todorov, I., & Sinha, M. (2021). Defining Research Software: a controversial discussion (Version 1). Zenodo. https://doi.org/10.5281/zenodo.5504016

# Goals

- ★ Introduction: why are we here?
- ★ Open Science isn’t complete without Open Source
- ★ **Guided exploration: into the Software Heritage archive**
- ★ Good practices for #BetterMetadata
- ★ The role of libraries: collaborating with OSPOs

# Source code is fragile!


**Google Kills Off Google Code**

Natasha Lomas @riptari / 10:58 AM GMT+1 • March 13, 2015

1.4 million projects

SD Times

**Google begins shutdown of its code repository**

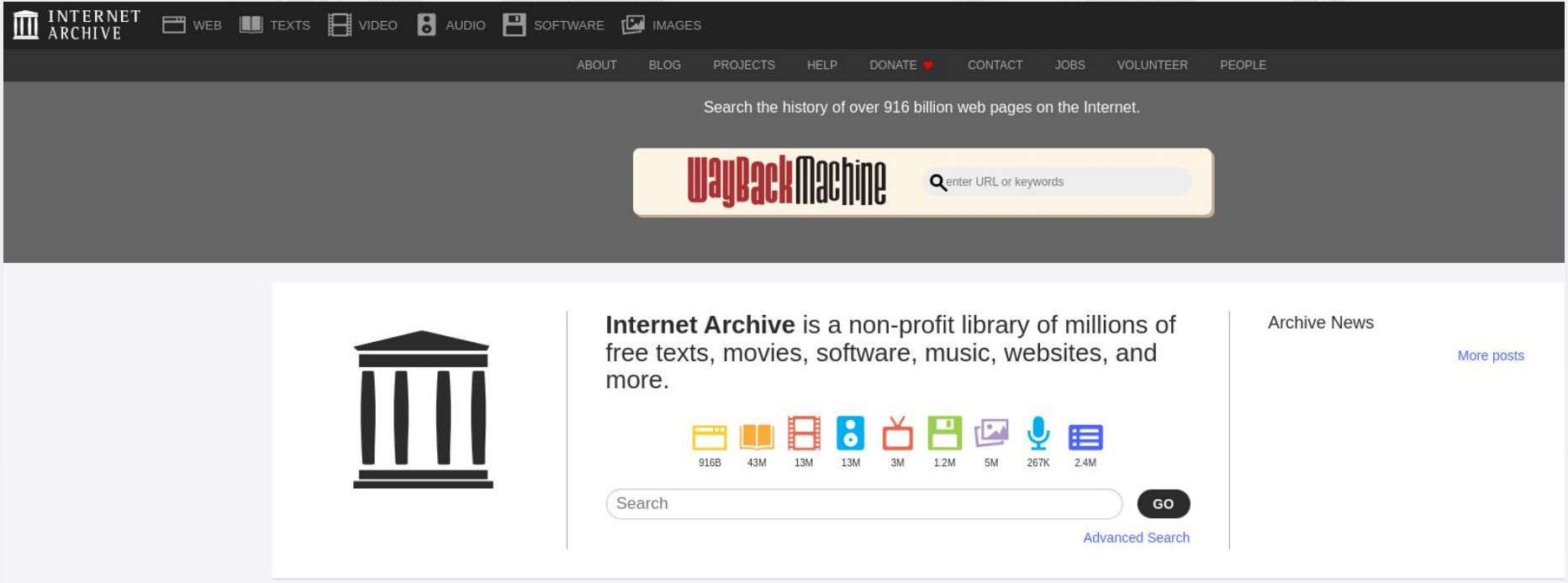
Latest News Published: March 12th, 2015 - Michael Pehel

After nine years, Google's open-source code repository, Google Code, started closing shop today by disabling new projects and announcing the permanent shut down of the service by January 26, 2016.


Google Code started as Google's answer to SourceForge, the predominant code repository back in 2006. The reliability of SourceForge was brought into question that year when SourceForge.net's database was hacked and user data was compromised. Problems continued the following summer in 2007 with a temporary service outage in August. The appeal of a repository backed by Google only increased as a result.

# ERROR 404

# Archiving the web: the wayback machine



# The development history is key

### Software evolves over time

➔ projects may last decades

➔ the development history is key to its understanding

### Complexity

- millions of lines of code
- large web of dependencies
- easy to break, difficult to maintain
- sophisticated developer communities

We need the complete snapshot

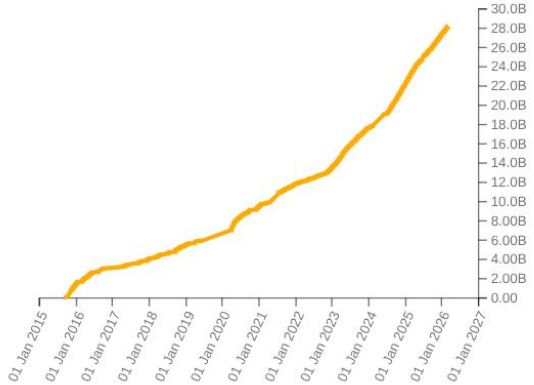
# A **universal** source code archive

**Collect, preserve and share all software source code**

Preserving our heritage, enabling better software and better science for all

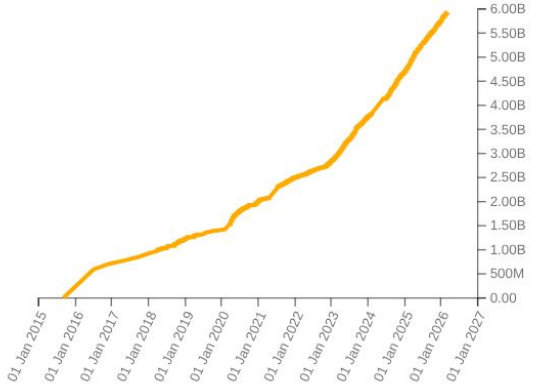
Source files

28,148,748,800



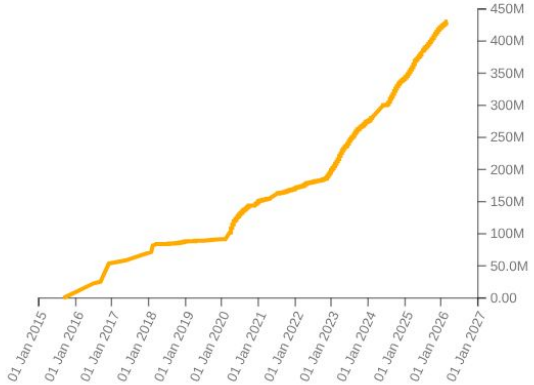

Commits

5,920,096,480




Projects

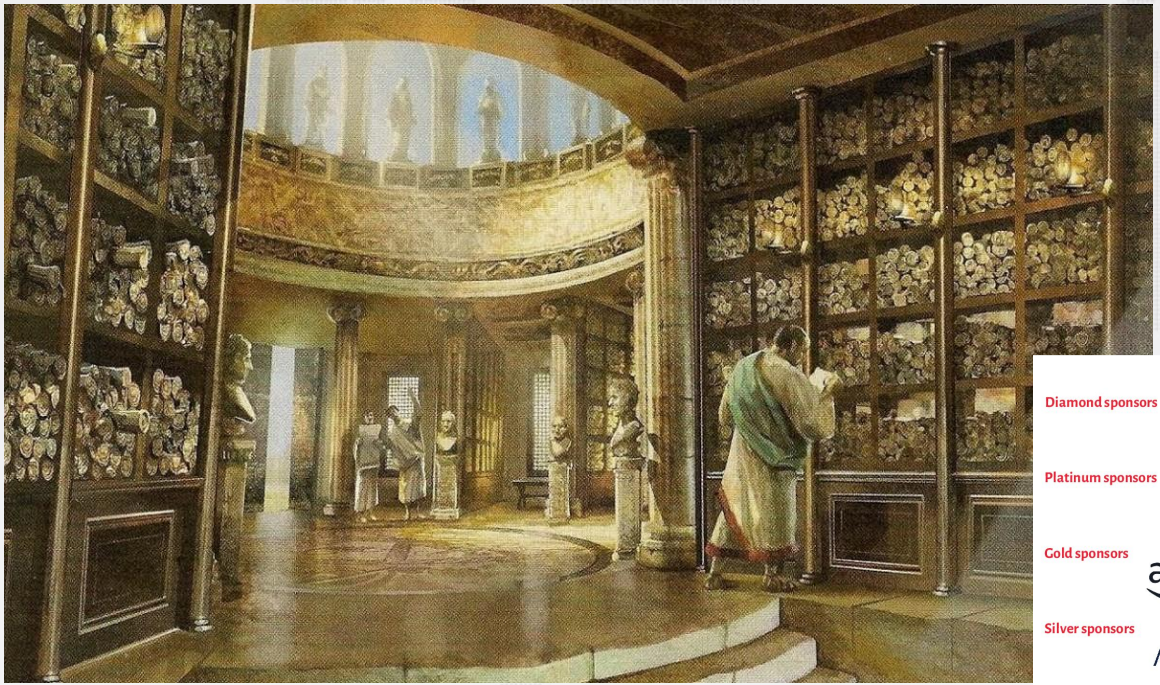
429,963,770




https://archive.softwareheritage.org/

Snapshot: March 2nd 2026

# The library of Alexandria of source code



Launched in 2016 by

Hosted SWH Foundation by

The 2025-2026 Sponsors

Diamond sponsors

Platinum sponsors

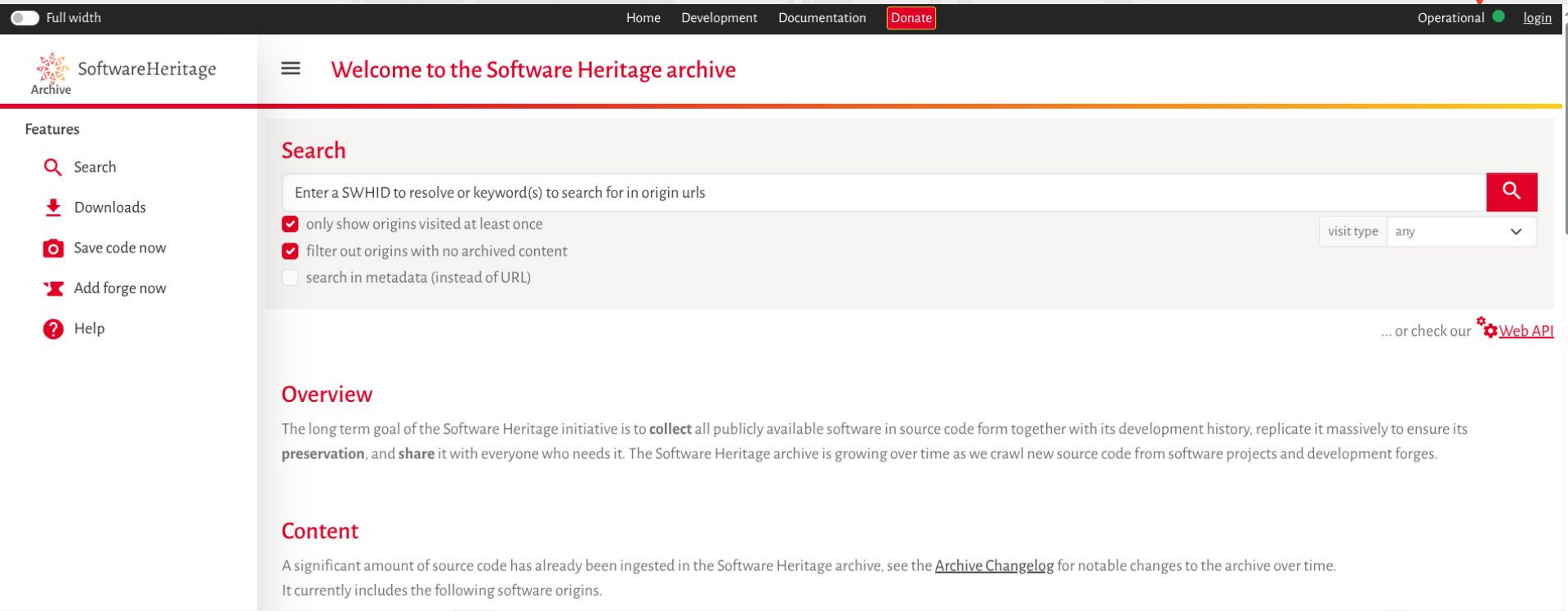
Gold sponsors

Silver sponsors

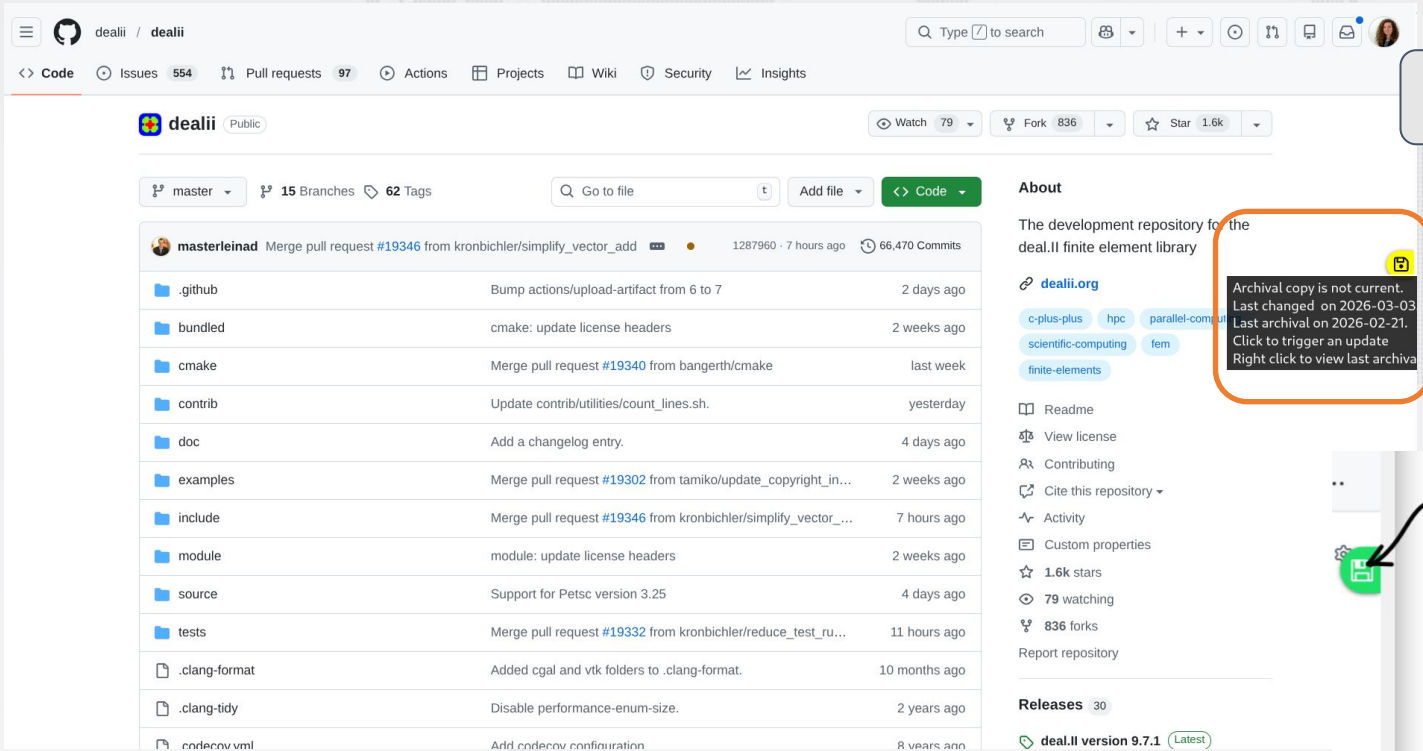
Bronze sponsors

In Collaboration with

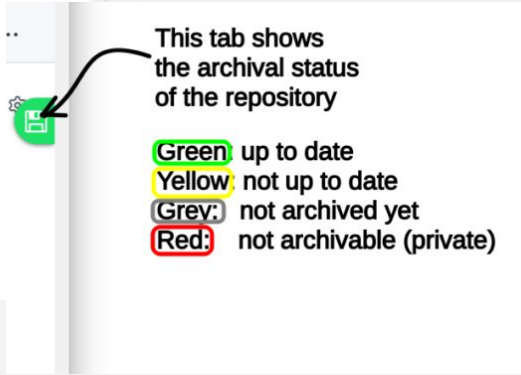
# Search for your code repository



# Archive without leaving the forge

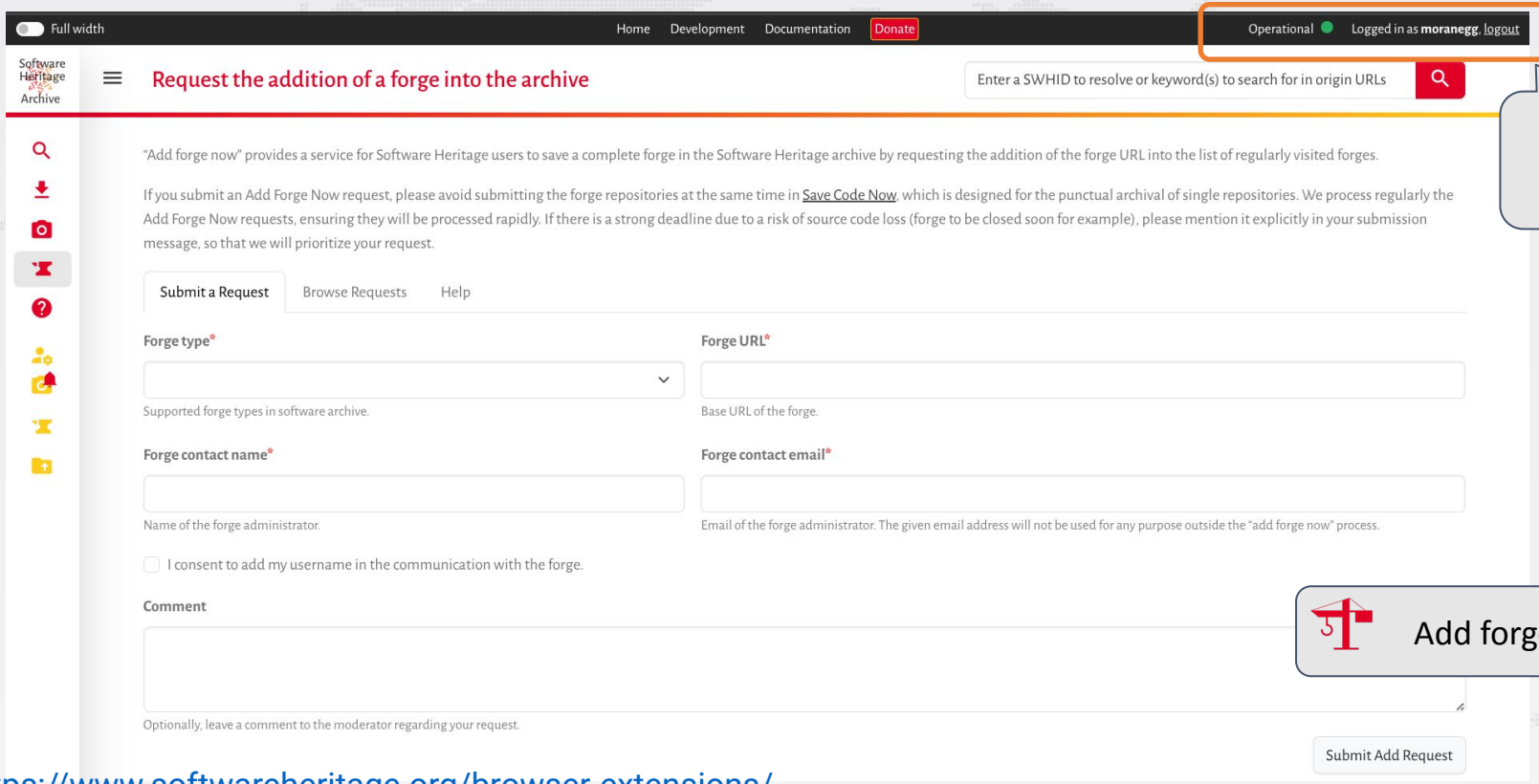



The Browser extension




https://www.softwareheritage.org/browser-extensions/

# Archive a complete forge instance

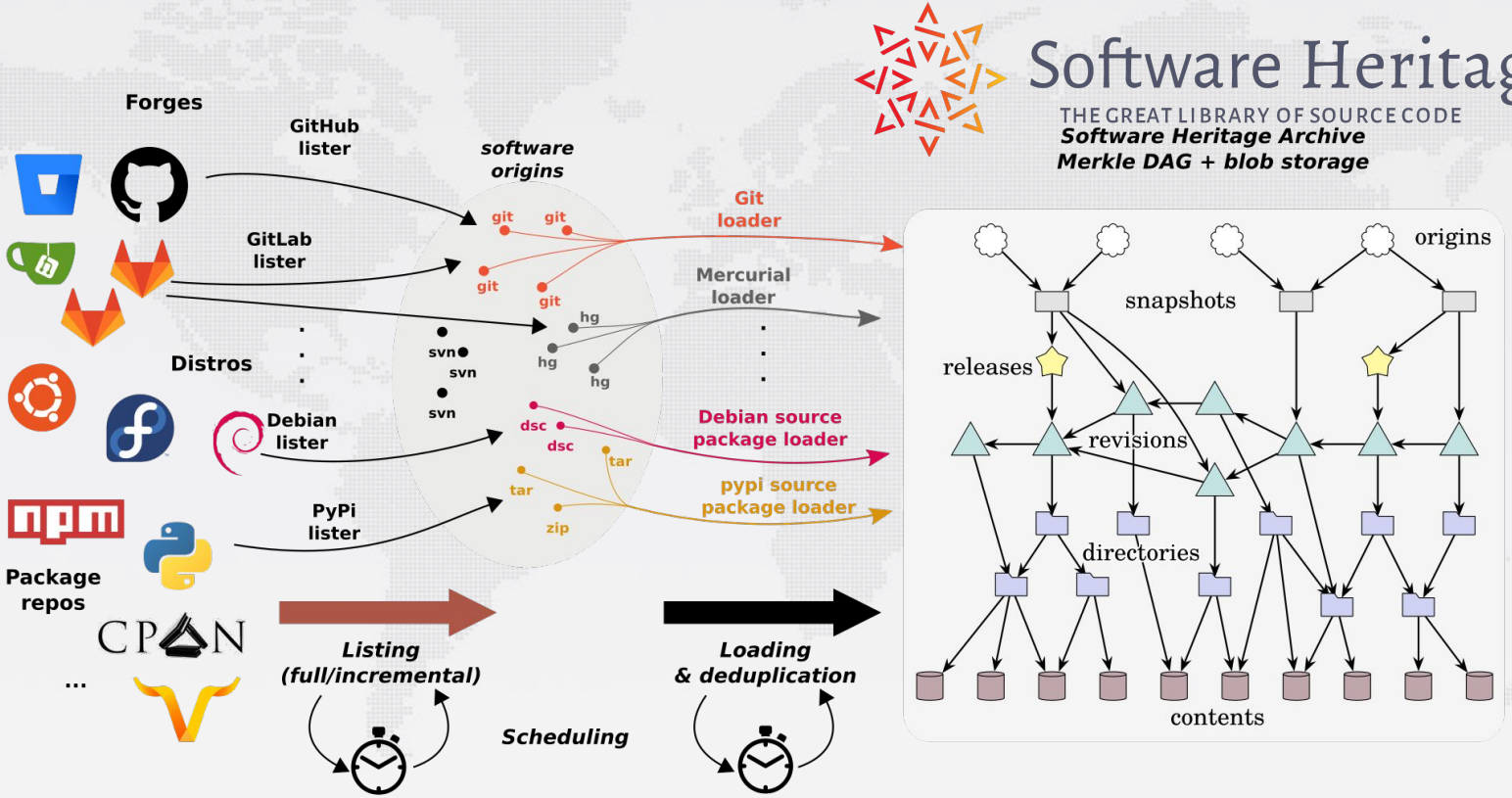



Must be logged in

Add forge request

https://www.softwareheritage.org/browser-extensions/

# A resilient solution: Preserving code from forges

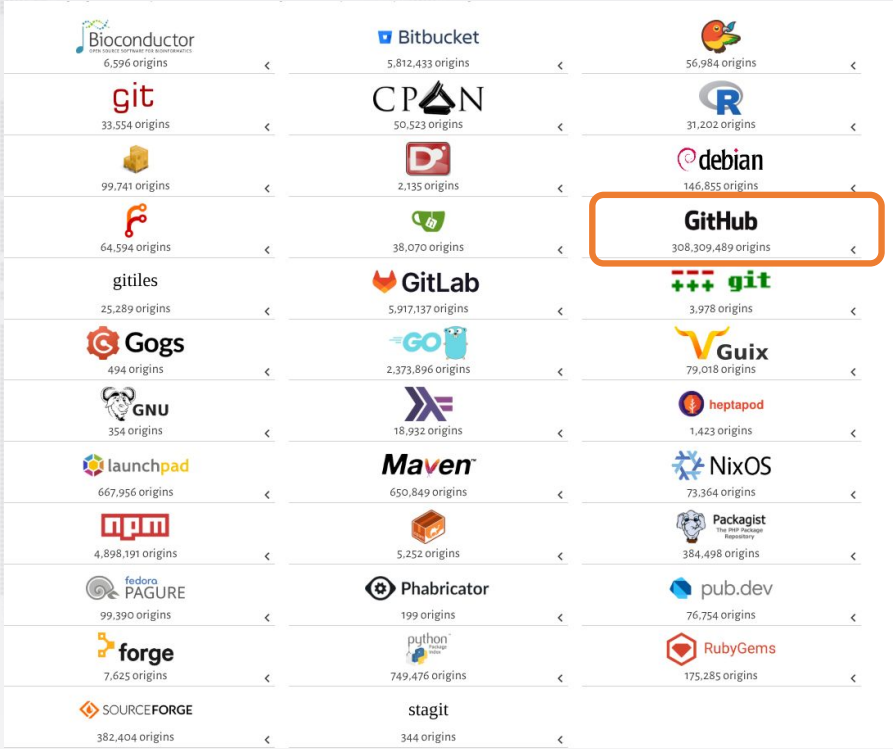


# Crawling: listing and loading

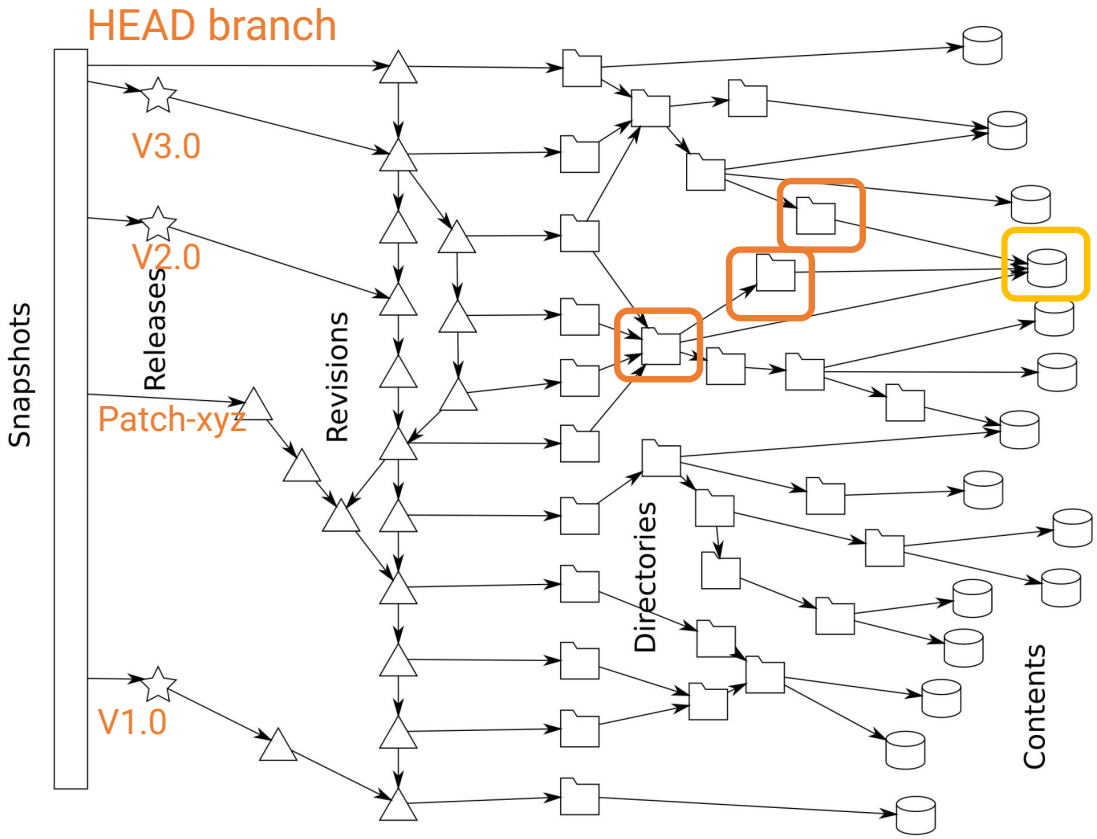
The SWH archive harvests actively source code from different sources and converts all the source code into a single and universal data structure which is an enormous Merkle DAG [Merkle, 1987].

Crawling is separated into three phases:

1. **listing** software sources,
2. **scheduling** updates and
3. **loading** the software artifacts into the archive.




Full development history permanently archived in a uniform data model




Repo github.com/namespace/repo-abc Visited on DD/MM/YYYY

No deduplication

# Granularity level does matter

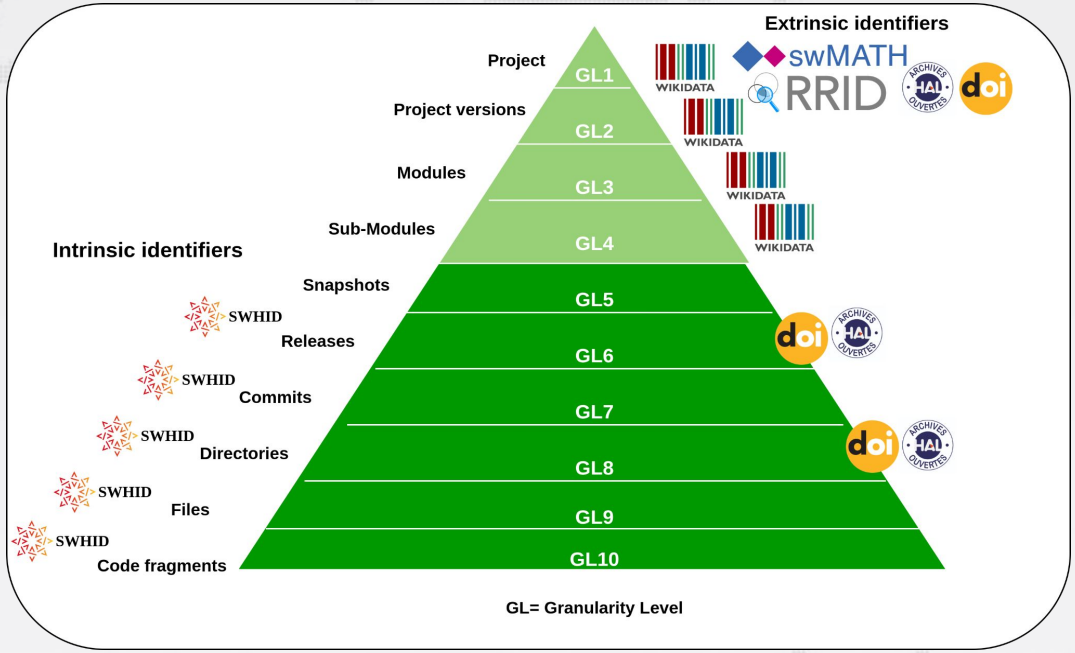
**Software concept / project / collection**

Description in registry, a homepage or any other form of metadata record

- ➔ Project versions (e.gPython2 and Python3)
- ➔ Modules
- ➔ Sub-modules

**Software artifact**

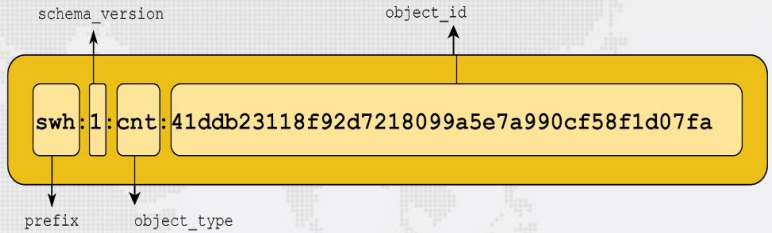
- ➔ Executable (download link of binary)
- ➔ Software source code
  - ◆ Dynamic artifact - living code repo
  - ◆ Archived copy
    - Snapshot (all branches, all dev history)
    - Release / Package
    - Commit- a specific point in development history
    - Directory
    - File
    - Algorithm

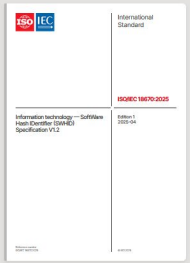


https://swhid.org/

# Introducing the SoftWare Hash Identifier

- ➔ **Intrinsic**: compute a unique digital fingerprint
- ➔ **decentralised**: do not need a registry, only agreement on a standard
- ➔ **cryptographically strong** identifiers



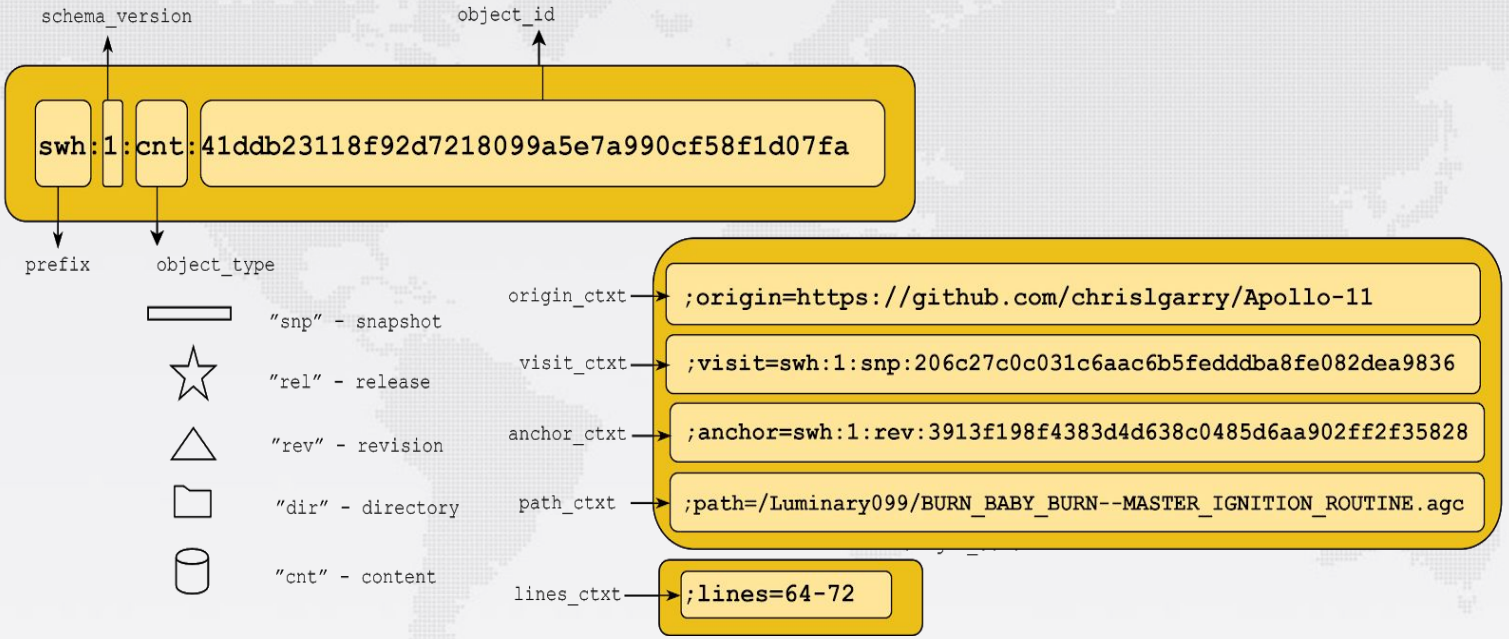
Read sample

## ISO/IEC 18670:2025

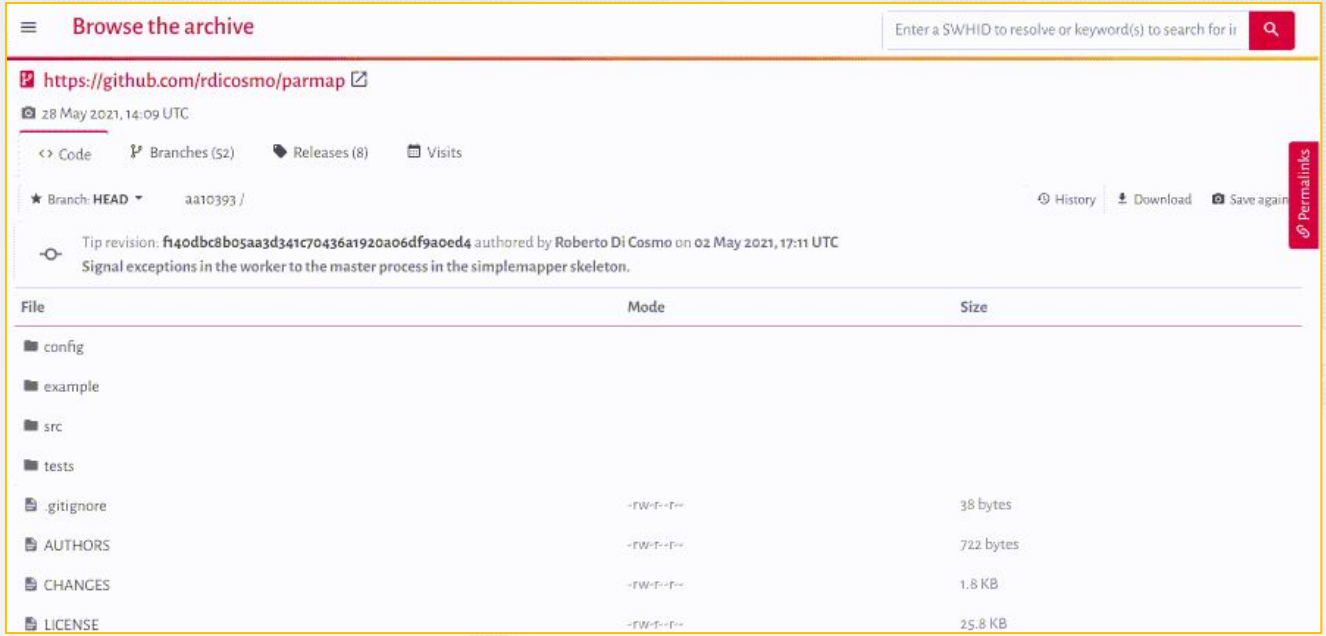
Information technology — SoftWare Hash IDentifier (SWHID) Specification V1.2

Published (Edition 1, 2025)

# SoftWare Hash Identifier with context



# Getting a SWHID is easy as ABC




Any end-user can retrieve a SWHID:

- no need to create an account on SWH
- no need to be the software author

# Extrinsic vs. Intrinsic

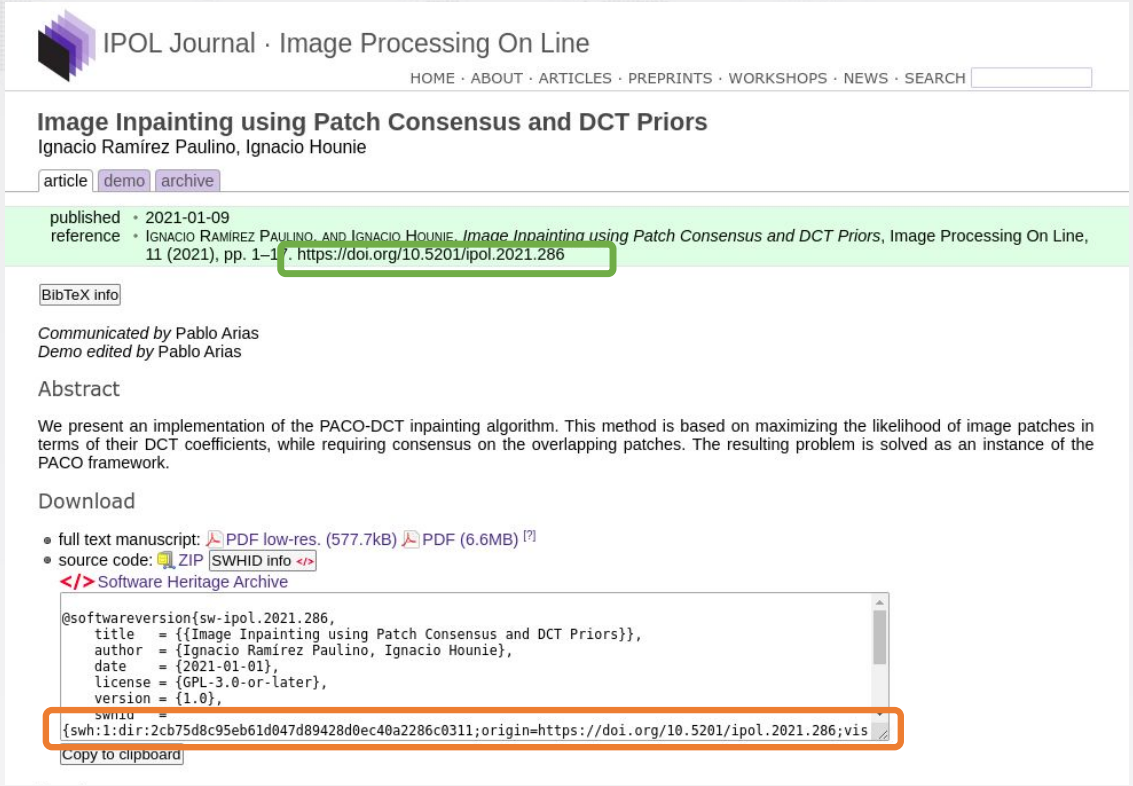
Main difference is how the relation between **identifier** and designated object is created and maintained.

Persistence is a key desired property.

| | Extrinsic | Intrinsic |
|---|---|---|
| Relation | Register | Convention - agreed protocol/naming |
| Pre-internet | Passport number, ISBN, SSN, etc. | Music / Chemistry notation |
| Internet era | DOI, Handle, ARK, RRID, etc. | Cryptographic hashes, e.g git, bitcoin, SWHID |

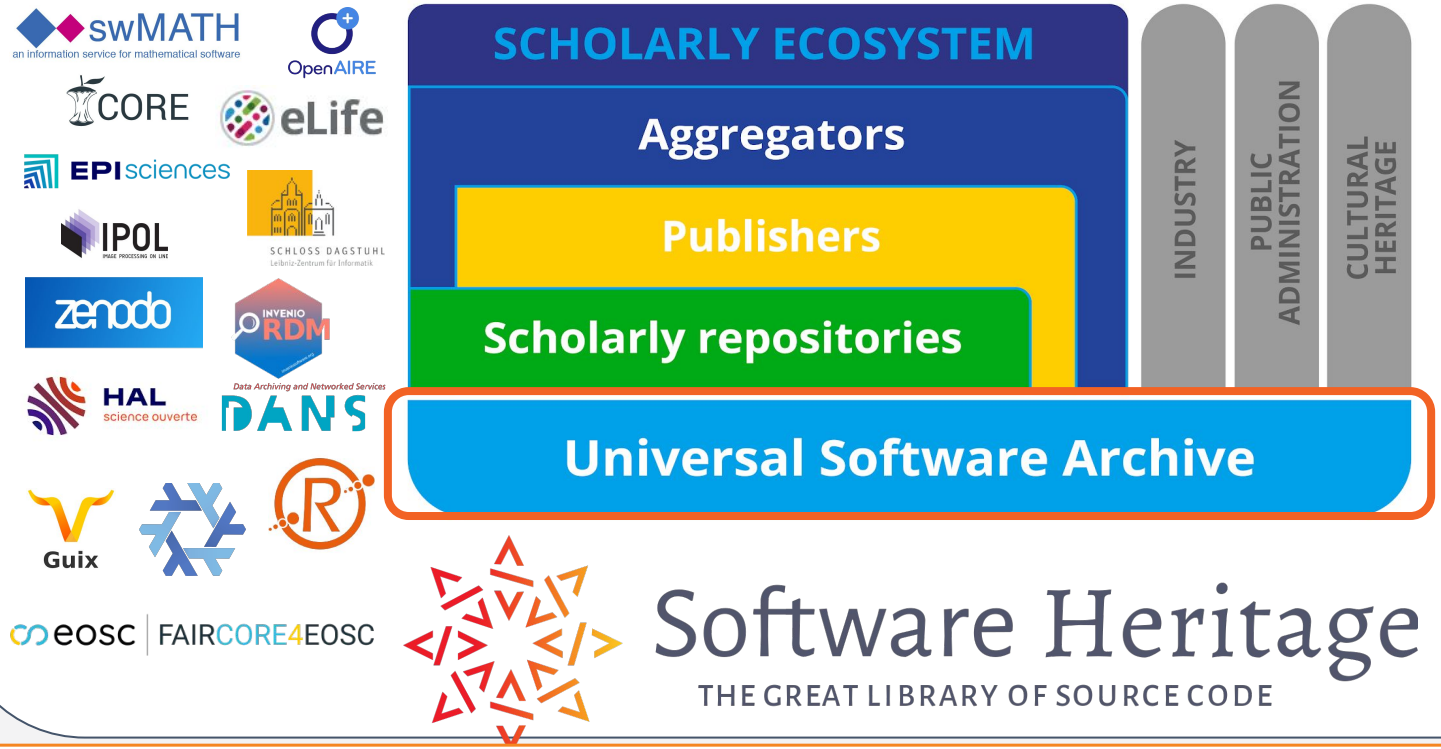
# A royal wedding: DOI x SWHID

- Journal Image Processing On Line (IPOL, https://www.ipol.im/)
- On Software Heritage
  - The article DOI is the **origin**: (https://doi.org/10.5201/ipol.2021.286)
  - The software SWHID: swh:1:dir:2cb75d8c95eb61d047d89428d0ec40a2286c0311



# Serving the scholarly ecosystem

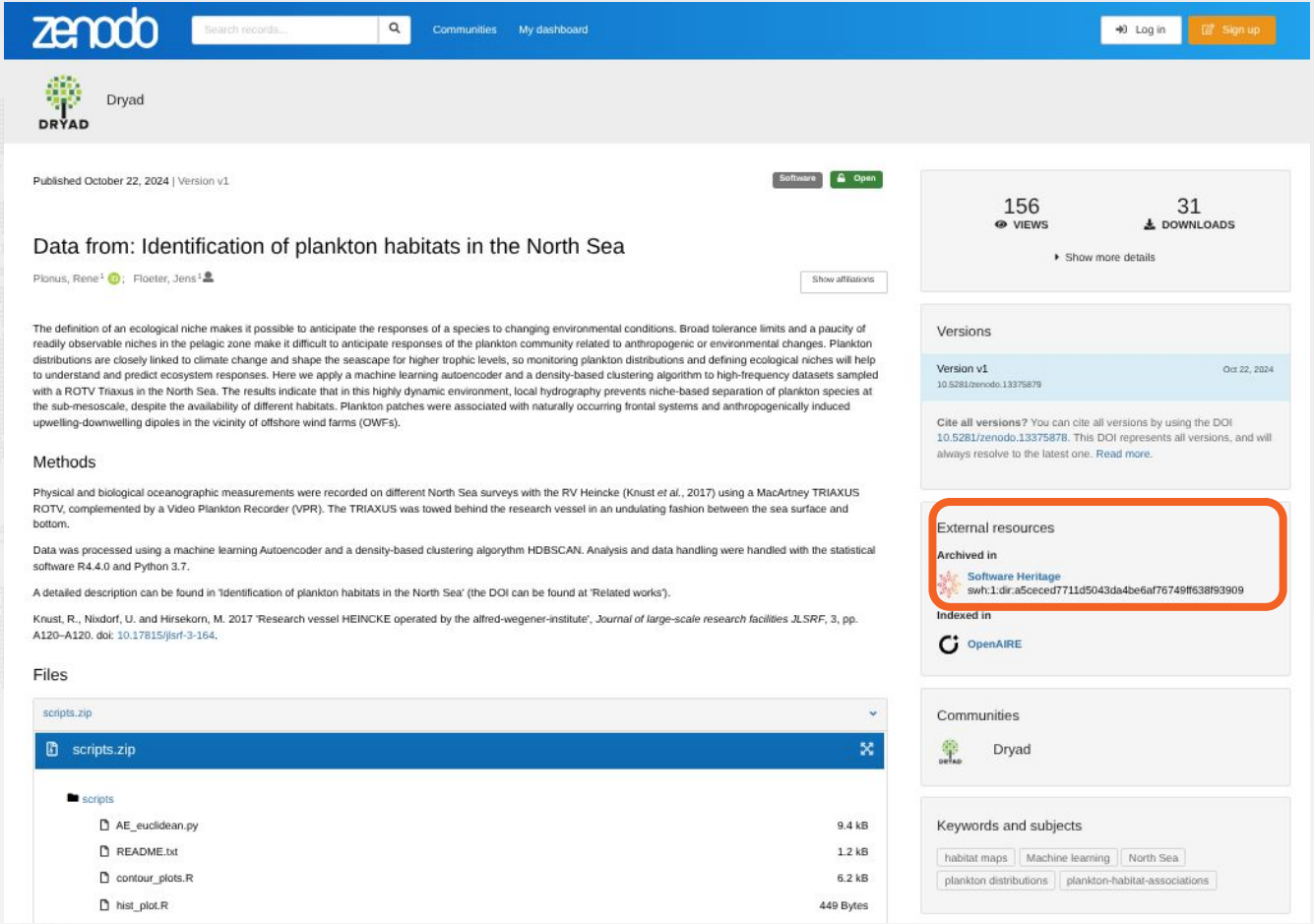
## Creating connections and mutualizing efforts




Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

# Software records are deposited and linked on **Zenodo**




eosc | FAIRCORE4EOSC

# Building for resilience

*“Let us save what remains: not by vaults and locks which fence them from the public eye and use in consigning them to* ***the waste of time****, but by such a* ***multiplication of copies****, as shall place them beyond the reach of* ***accident****.”*

— Thomas Jefferson

# The Software Heritage Mirrors Network: A distributed multi stakeholders infrastructure

# Goals

- ★ Introduction: why are we here?
- ★ Open Science isn’t complete without Open Source
- ★ Guided exploration: into the Software Heritage archive
- ★ **Good practices for #BetterMetadata**
- ★ The role of libraries: collaborating with OSPOs

# What is at stake?

#DigitalSovereignty

**Archive**

➔ make sure we can access to retrieve the software (reproducibility)

**Reference**

➔ make sure we can identify the software artifacts (reproducibility)

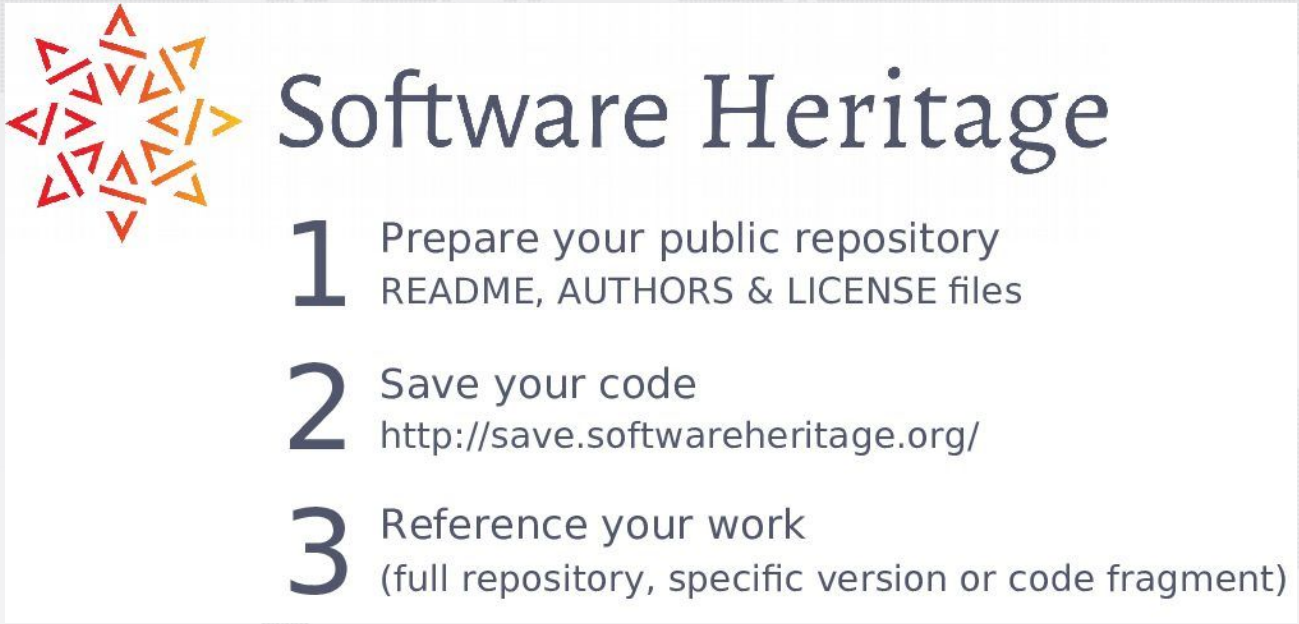
**Cite (for credit)**

➔ make it rewarding to create software by giving credit to authors (evaluation!)

**Describe**

➔ make it easy to discover the software projects (visibility)

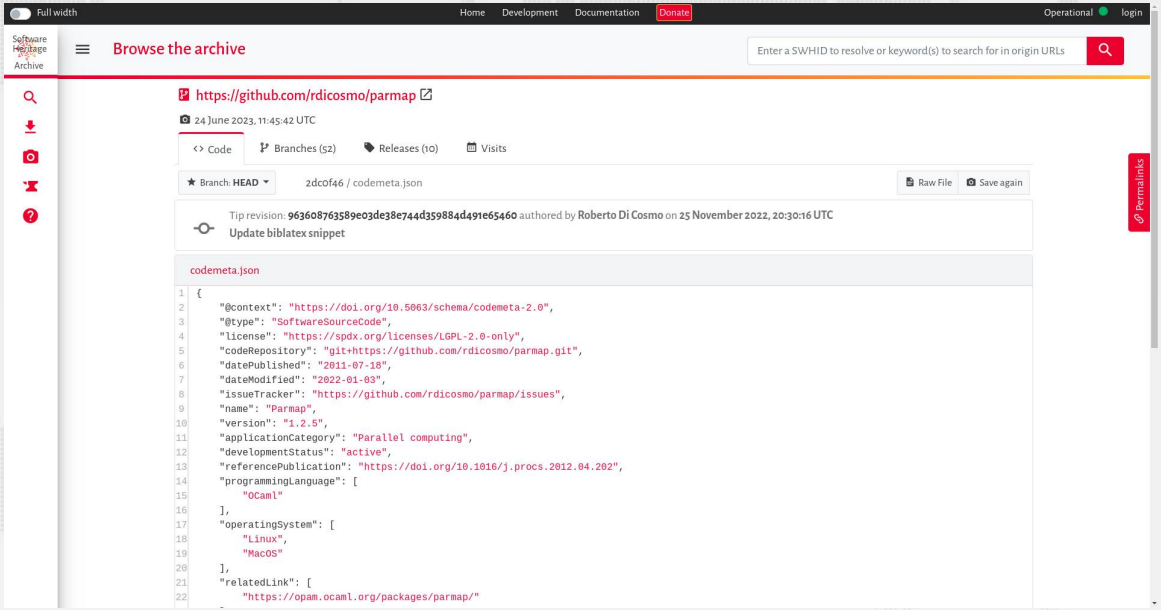
# **Sharing**, **curating** and **archiving** your source code



# **Step #1**: Prepare code with metadata

In the *software source code* itself

- README
- LICENSE
- AUTHORS
- codemeta.json
- package management
  - pom.xml
  - package.json
  - …
- CITATION.cff
- .About
- …



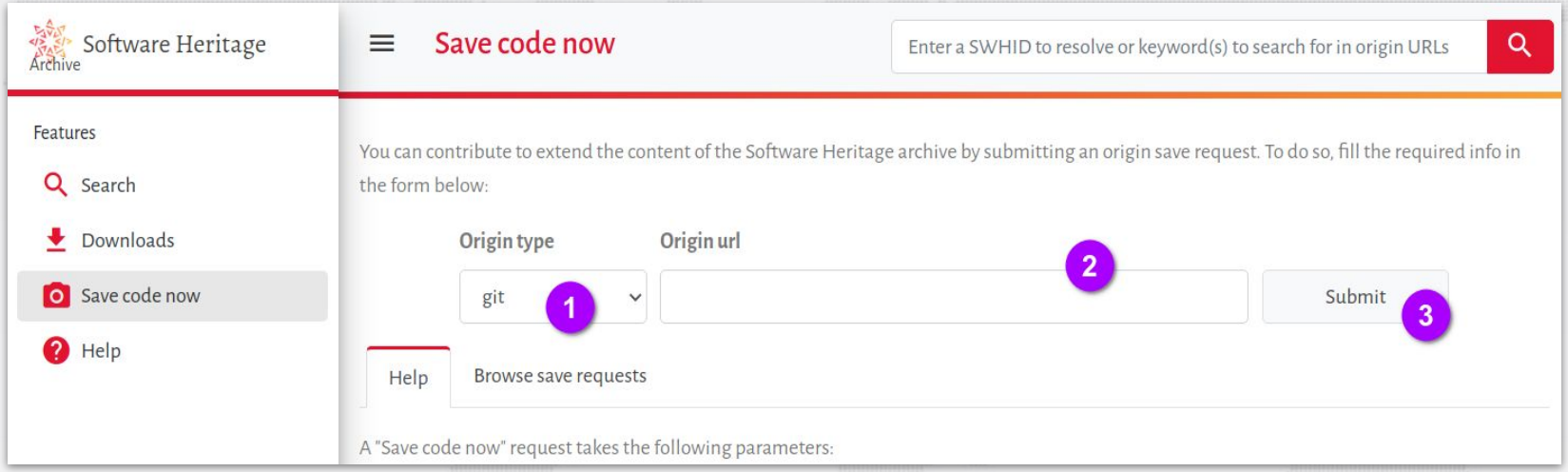

**Human readable (e.g README)**

**Machine actionable (e.g codemeta.json)**

# Step #2: Save ~~your~~ any code to SWH

https://save.softwareheritage.org/

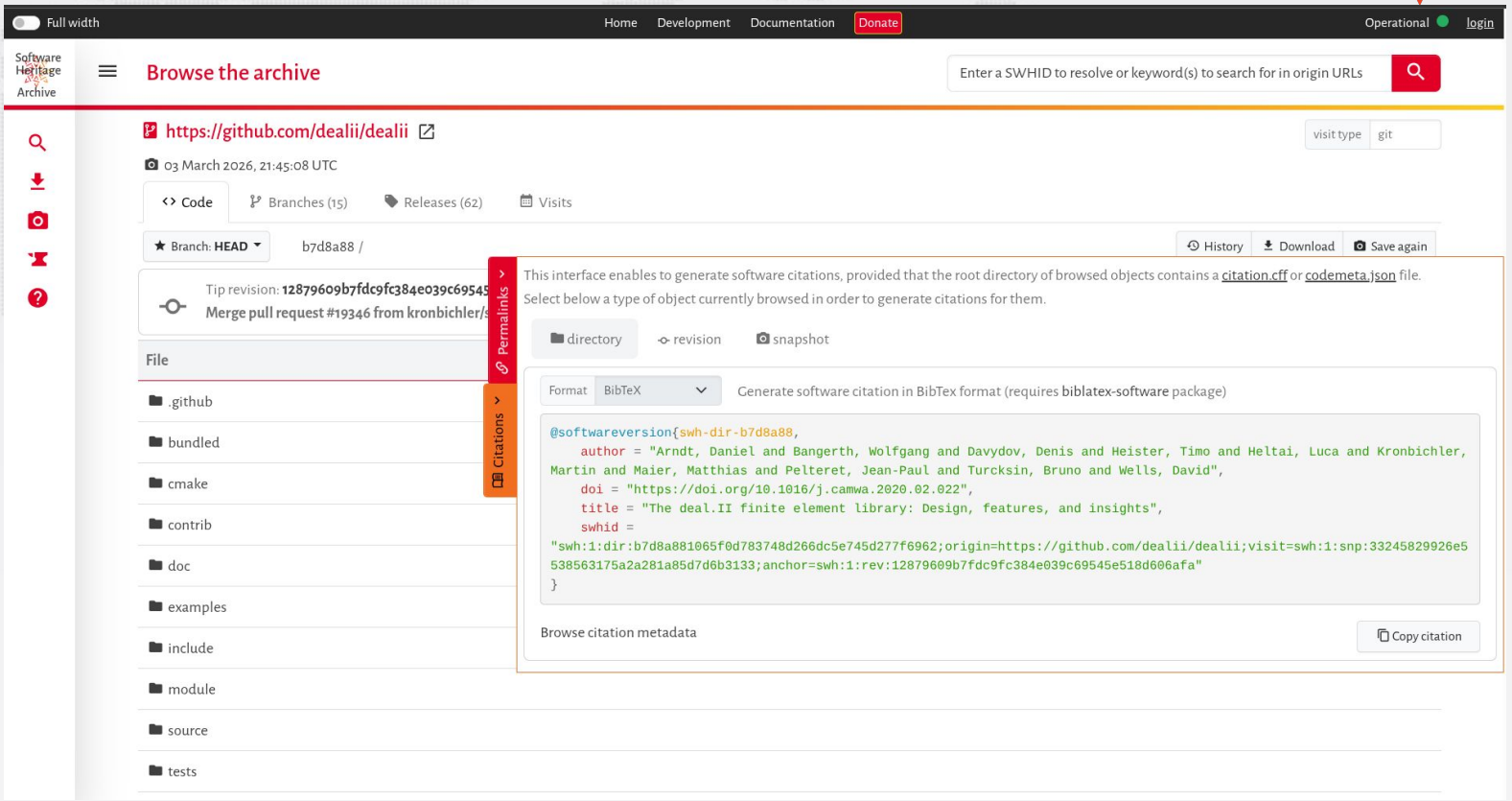
**Save code now**




**Documentation**

https://www.softwareheritage.org/how-to-archive-reference-code/

# **Step #3**: Reference and cite code with the SWHID



# The **R**esearch **S**oftware **M**eta**D**ata guidelines

**SIRS report** **FAIR4RS**

**The RSMD seven <u>Aspects</u>**

1. **General Metadata Requirements**
2. **Accessibility & preservation** — A = Archive — A = Accessible
3. **Reference & identification** — R = Reference
4. **Description & classification** — D = Describe — F = Findable — I = Interoperable
5. **Attribution & credit** — C = Cite
6. **Reuse, licensing & legal aspects** — R = Reusable
7. **Re-execute: Dependencies & execution environment**

https://github.com/FAIR-IMPACT/RSMD-guidelines
https://fair-impact.github.io/RSMD-guidelines/

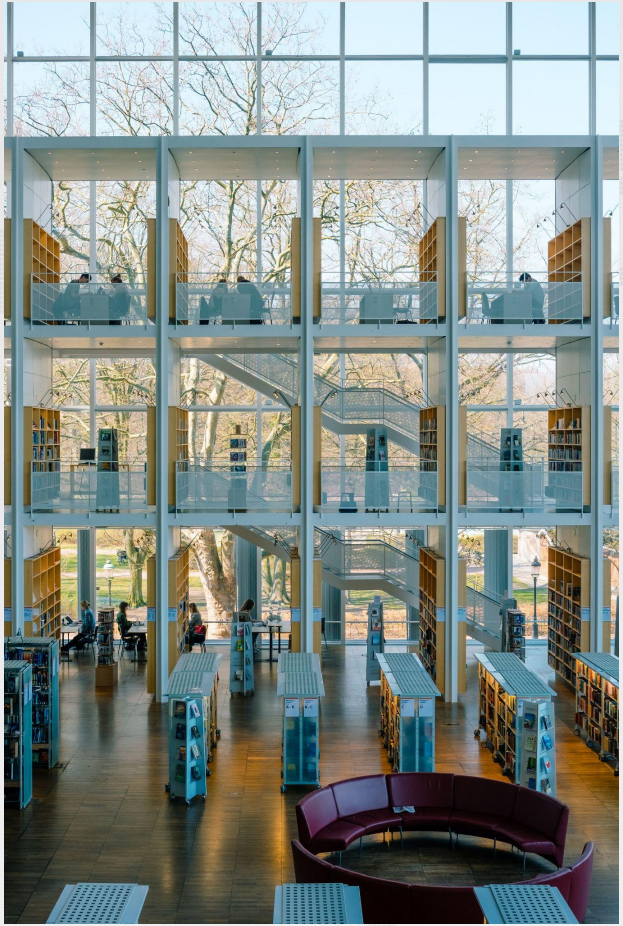
eosc | FAIR-IMPACT

# Questions sur les partenaires

Sur menti:  6268 6204

# Goals

- ★ Introduction: why are we here?
- ★ Open Science isn’t complete without Open Source
- ★ Guided exploration: into the Software Heritage archive
- ★ Good practices for #BetterMetadata
- ★ **The role of libraries: collaborating with OSPOs**



Picture by Ciprian Boiciuc on Unsplash

# Research libraries have a pivotal role

Software preservation requires a global and coordinated effort.

The long road ahead:

- availability,
- findability,
- traceability,
- reproducibility

# Supporting the supporters: The **A**rchives and **L**ibraries **I**nterest **G**roup (**ALIG**)

## Open infrastructure
Contribute to a 100% Open Source Infrastructure, valuing users input and support

## Tailored Support
Receive assistance for your software archival questions, plans and projects

## Community
Participate in an inclusive, innovation driven community

## *ALIG Members*

Sant'Anna School of Advanced Studies – Pisa

Canadian Research Knowledge Network
Réseau canadien de documentation pour la recherche

université PARIS-SACLAY

Konsortium der Schweizer Hochschulbibliotheken
Consortium des bibliothèques universitaires suisses
Consorzio delle biblioteche universitarie svizzere
Consortium of Swiss Academic Libraries

couperin.org
Consortium unifié des établissements universitaires et de recherche pour l'accès aux publications numériques

DANS

https://www.softwareheritage.org/support/members-alig/

# Who are the stakeholders?

## Researchers

- find and access useful software
- get credit for software
- verify/reproduce results

## Laboratories/teams

- track contributions
- produce reports
- maintain web pages

## Research Organization

know its software assets for:

- technology transfer,
- impact metrics,
- strategy

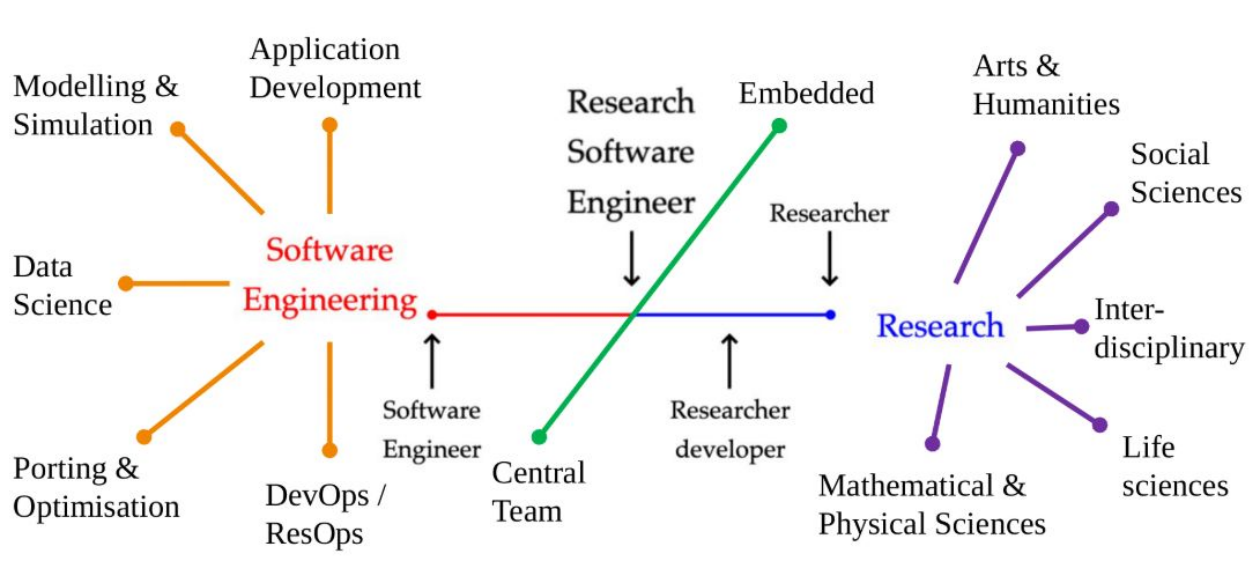
## Curators

- verify and curate metadata
- provide documentation on software curation
- train researchers for quality metadata

# What about **R**esearch **S**oftware **E**ngineers?

## RSEs

- central role for creating and maintaining Research Software
- building bridges between Academia and Software Ecosystems
- support teams with technology




Chue Hong, Neil (2023). Is Research Software Engineering coming of age?. figshare. Presentation. https://doi.org/10.6084/m9.figshare.24078054.v5

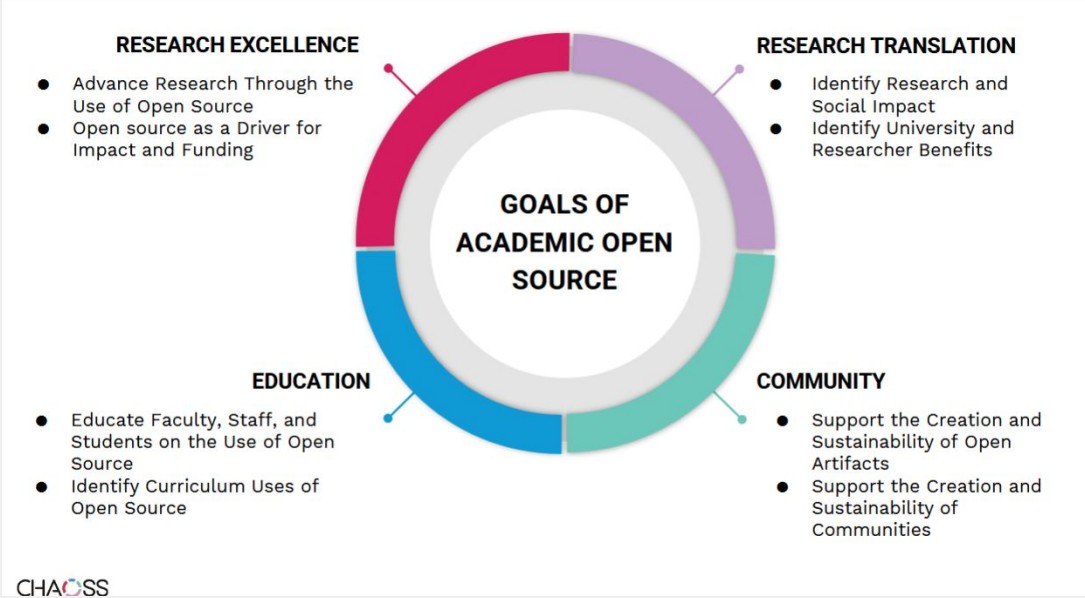
# OSPO for Open Source Programme Office

https://code.gouv.fr/fr/blog/definition-ospo/

**OSPO**

- central role for Open Source strategy
- building bridges between Academia and Open Source
- support teams with technology

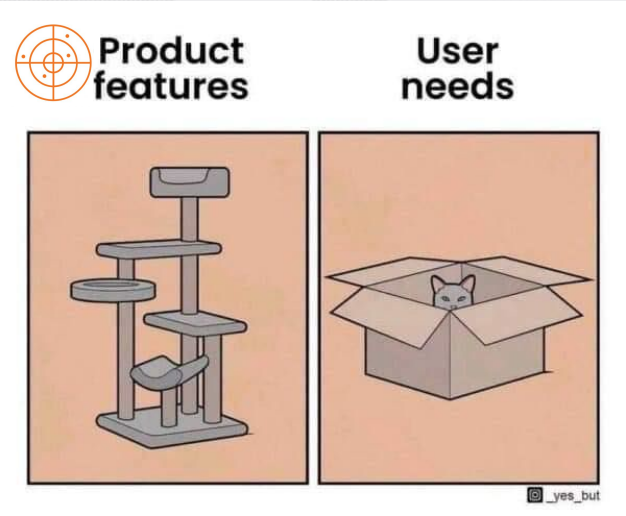
An Open Source Program Office (OSPO) is an entity within an institution that **defines and implements an open source strategy.**




(C. Dillon, 2025)

Personas full description

# **OSPO-RADAR**: A layer on top of the SWH archive




*“Open-source software development is an essential piece of the science ecosystem that drives and sustains scientific discovery,”* said **Joshua M. Greenberg**, Program Director, Sloan Foundation.
*“Software Heritage’s OSPO-RADAR project will make visible that influence by giving research institutions a set of tools to better manage, document, and share their open-source work.”*

ALFRED P. SLOAN FOUNDATION

# Designing a product with the OSPOs Community

The MVP capabilities

- Create an **institution collection**
  - **Batch import;**
    - **Signal code repo**
- **Management dashboard**: to view/filter/curate;
- A public **front-shop** that can be injected in institutional website through an **API**.


**Funded by**

ALFRED P. SLOAN FOUNDATION

[Full specifications](#) are public

# Software Heritage as a Metadata-broker

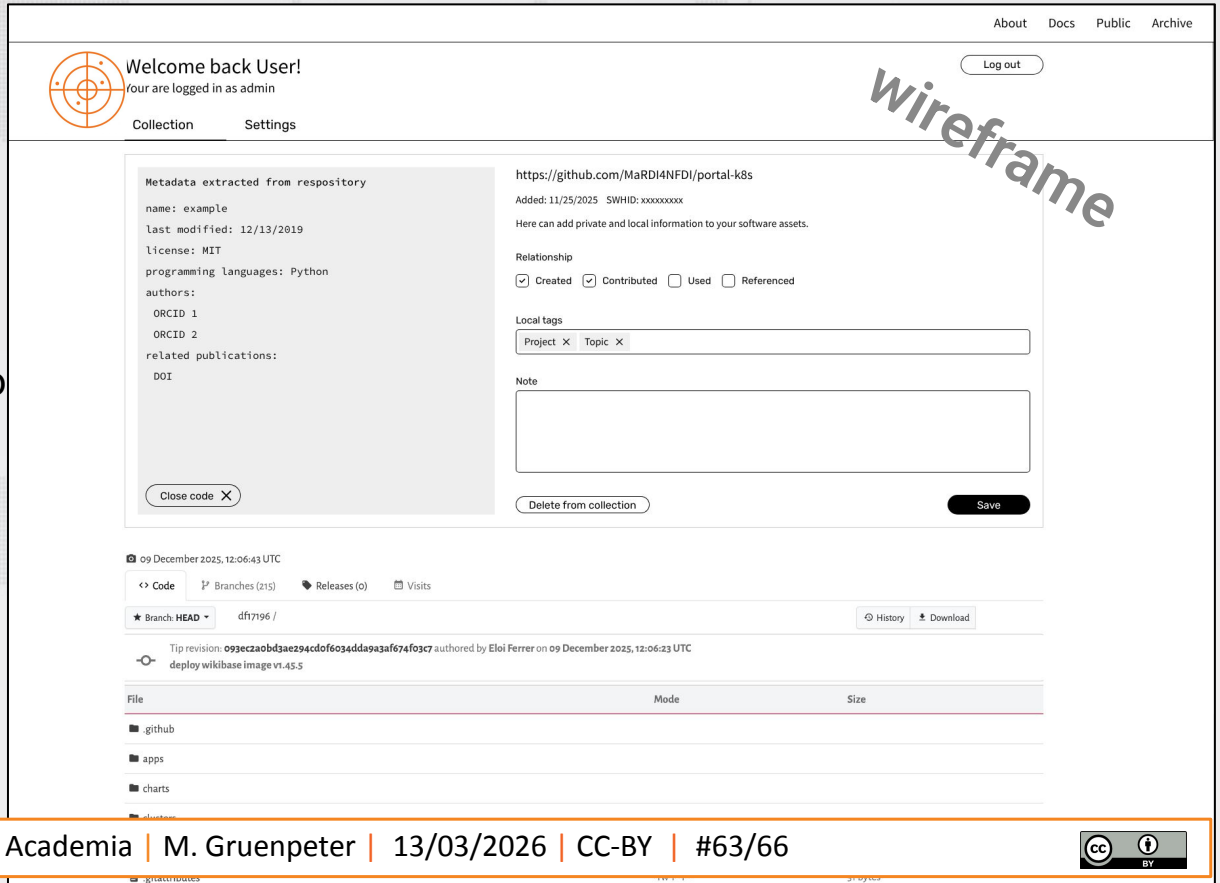
Leveraging the CodeMeta vocabulary

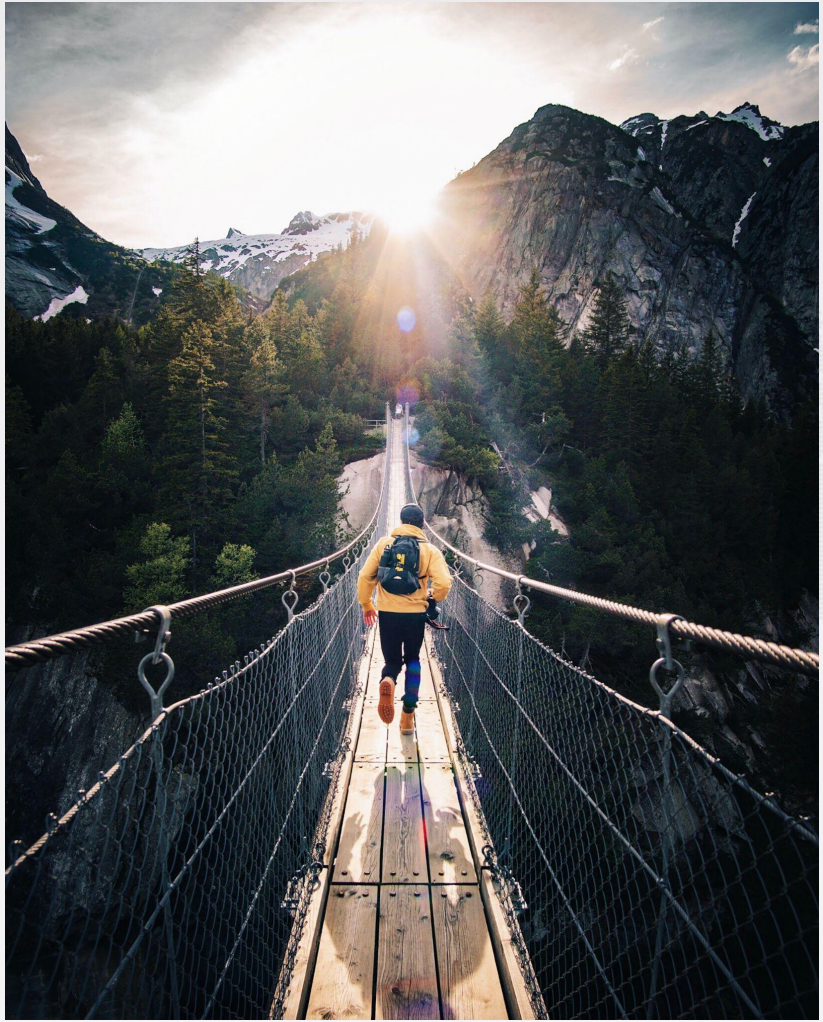
- A subset of **schema.org**
- Community governed project

The CodeMeta Project

Software Heritage has the capacity to

- Ingest metadata
- Extract metadata
- Deliver consolidated views to stakeholders





Picture by Fabio Comparelli on Unsplash

**RSEs** and **OSPOs** in Academia can support **Librarians**

And connect *Academia* to the rest of the *software world*

Join the mailing-list to follow on news on new features for OSPOs

- swh-ospo-radar@inria.fr

# Building together interoperable Ecosystems

Software source code recognition requires a **supported** and **coordinated** effort. We need to treat code not just as a tool, but as a core pillar of our intellectual heritage.

SWHID

Mutualisation

Recognition

Sustainability

Curation &
Quality metadata

Open by default

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

# Thank you

morane@softwareheritage.org
https://www.softwareheritage.org/

Browse the **source code archive**
https://archive.softwareheritage.org/

Become a **member**
https://www.softwareheritage.org/support/members-alig

Volunteer as an **ambassador**
https://www.softwareheritage.org/ambassadors/

Subscribe to the **newsletter**
https://www.softwareheritage.org/newsletter/

The Team

© Inria / Photo B. Fourrier