eosc | FAIR-IMPACT
Expanding FAIR solutions across EOSC

Software as a first class output in the scholarly ecosystem

Morane Gruenpeter
Software Heritage
Inria, France

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Funded by the European Union

eosc | FAIRCORE4EOSC

Outline

#ResearchSoftware

A pillar of **Open Science**

Goal: Make software a first class output

Archive

Use cases

Reference vs. Citation

What about the metadata?

Software Heritage

Recognize software as a special research object

From Data vs. Software:

*Software is **executable** data is not.*

*Software is a **creative work**, scientific data are facts or observations*

2016: Katz DS et al. 2016. **Software vs. data** in the context of citation. PeerJ Preprints 4:e2630v1 https://doi.org/10.7287/peerj.preprints.2630v1

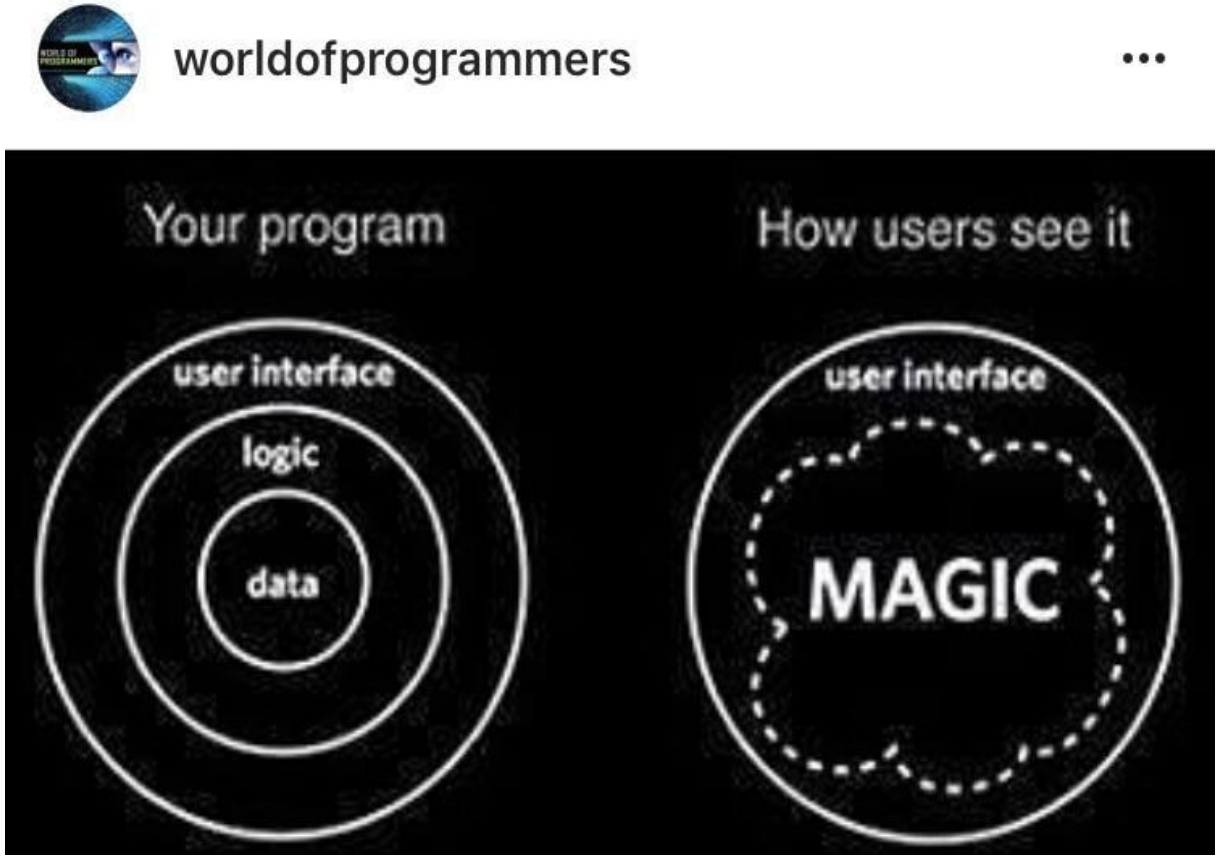
Opportunity note- Recommendation n° 2 :

*Make sure **the specific nature of software** is recognized and not considered as "just data" particularly in the context of discussion about the notion of FAIR data.*

2019: the **Opportunity Note** by the French national Committee for Open Science's Free Software and Open Source Project Group (Clément-Fontaine, 2019)

Software Heritage

What is Software?



Software as a concept

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

Not a digital artifact

Software artifacts

- Executables
- Source code

A very large collection of digital artifacts

Software Heritage

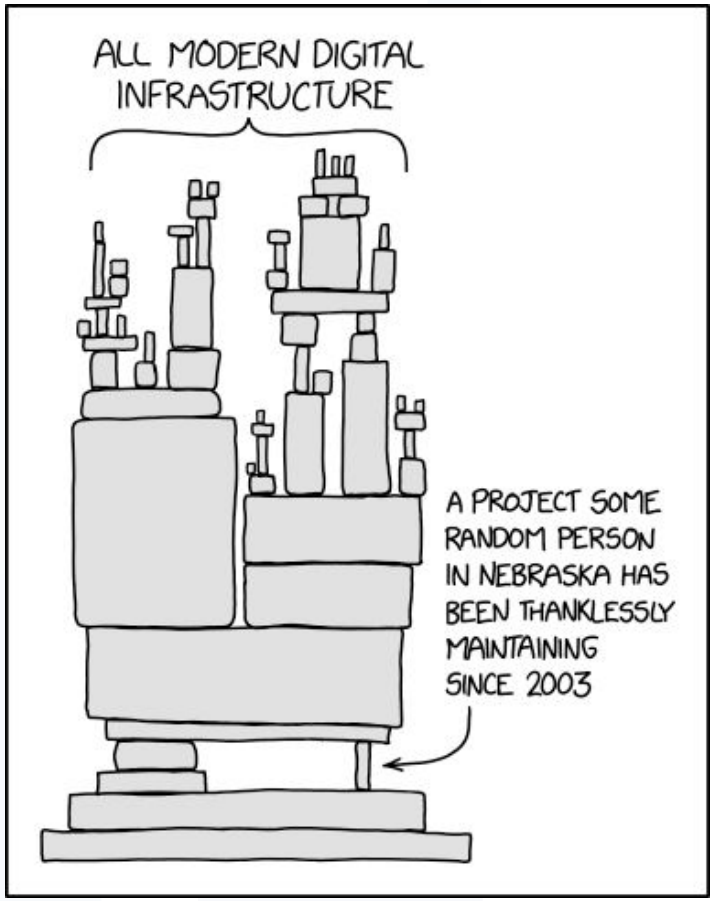
Software Source Code is special (not just data)

Software evolves over time

- projects may last decades
- the development history is key to its understanding

Complexity

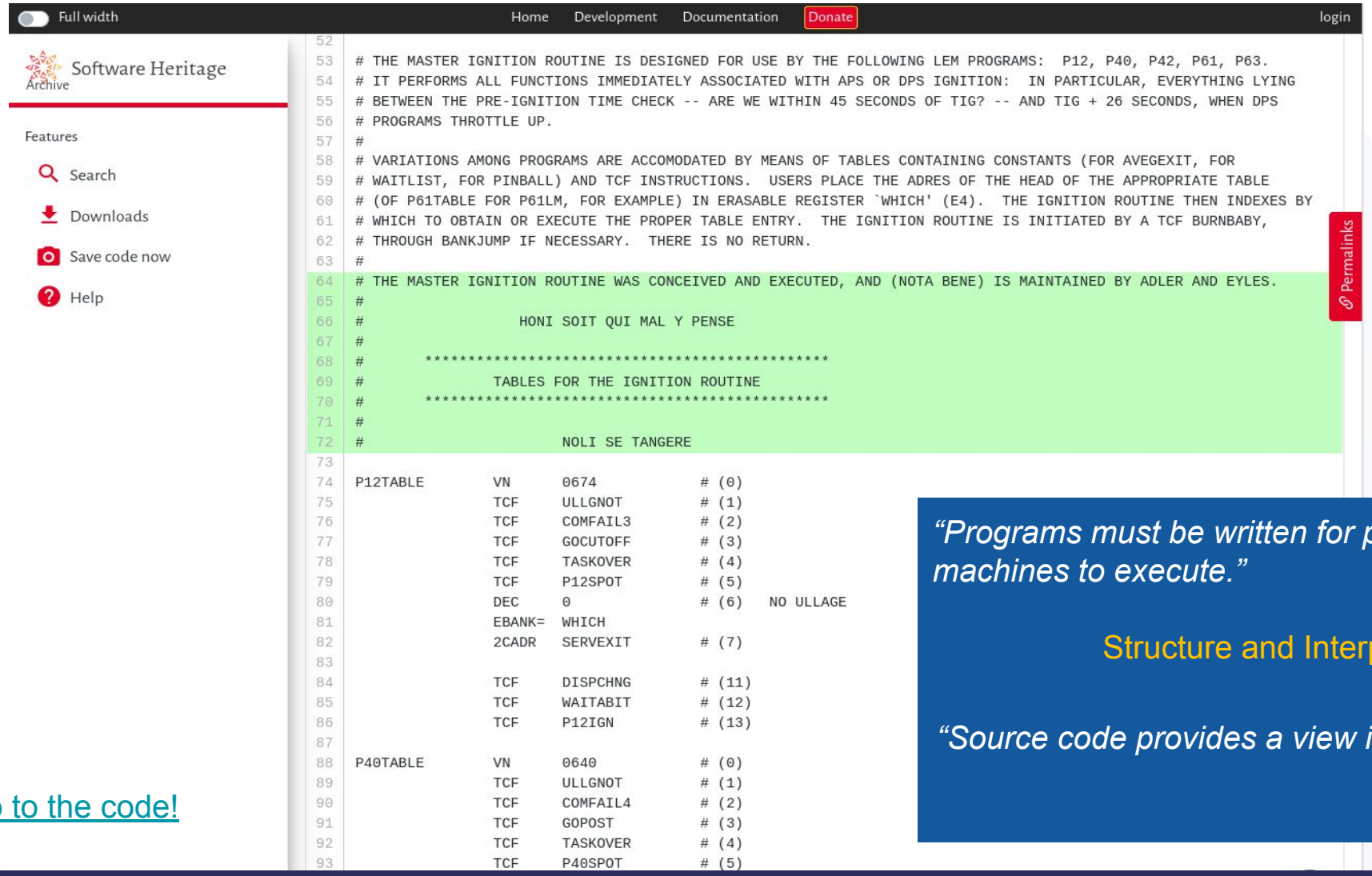
- millions of lines of code
- large web of dependencies
  - easy to break, difficult to maintain
- sophisticated developer communities



https://www.reddit.com/r/ProgrammerHumor/comments/ic1zmc/dependency_xkcd/

Software Heritage

Software Source Code human readable and executable knowledge




```
# THE MASTER IGNITION ROUTINE IS DESIGNED FOR USE BY THE FOLLOWING LEM PROGRAMS:  P12, P40, P42, P61, P63.
# IT PERFORMS ALL FUNCTIONS IMMEDIATELY ASSOCIATED WITH APS OR DPS IGNITION:  IN PARTICULAR, EVERYTHING LYING
# BETWEEN THE PRE-IGNITION TIME CHECK -- ARE WE WITHIN 45 SECONDS OF TIG? -- AND TIG + 26 SECONDS, WHEN DPS
# PROGRAMS THROTTLE UP.
#
# VARIATIONS AMONG PROGRAMS ARE ACCOMODATED BY MEANS OF TABLES CONTAINING CONSTANTS (FOR AVEGEXIT, FOR
# WAITLIST, FOR PINBALL) AND TCF INSTRUCTIONS.  USERS PLACE THE ADRES OF THE HEAD OF THE APPROPRIATE TABLE
# (OF P61TABLE FOR P61LM, FOR EXAMPLE) IN ERASABLE REGISTER `WHICH' (E4).  THE IGNITION ROUTINE THEN INDEXES BY
# WHICH TO OBTAIN OR EXECUTE THE PROPER TABLE ENTRY.  THE IGNITION ROUTINE IS INITIATED BY A TCF BURNBABY,
# THROUGH BANKJUMP IF NECESSARY.  THERE IS NO RETURN.
#
# THE MASTER IGNITION ROUTINE WAS CONCEIVED AND EXECUTED, AND (NOTA BENE) IS MAINTAINED BY ADLER AND EYLES.
#
#                  HONI SOIT QUI MAL Y PENSE
#
#       ***********************************************
#              TABLES FOR THE IGNITION ROUTINE
#       ***********************************************
#
#                     NOLI SE TANGERE

P12TABLE        VN      0674            # (0)
                TCF     ULLGNOT         # (1)
                TCF     COMFAIL3        # (2)
                TCF     GOCUTOFF        # (3)
                TCF     TASKOVER        # (4)
                TCF     P12SPOT         # (5)
                DEC     0               # (6)   NO ULLAGE
                EBANK=  WHICH
                2CADR   SERVEXIT        # (7)

                TCF     DISPCHNG        # (11)
                TCF     WAITABIT        # (12)
                TCF     P12IGN          # (13)

P40TABLE        VN      0640            # (0)
                TCF     ULLGNOT         # (1)
                TCF     COMFAIL4        # (2)
                TCF     GOPOST          # (3)
                TCF     TASKOVER        # (4)
                TCF     P40SPOT         # (5)
```

[Go to the code!](#)

"Programs must be written for people to read, and only incidentally for machines to execute."

Harold Abelson, 1985
Structure and Interpretation of Computer Programs (1st ed.),

"Source code provides a view into the mind of the designer."

Len Shustek, 2006
Computer History Museum

Defining Research Software

Research Software

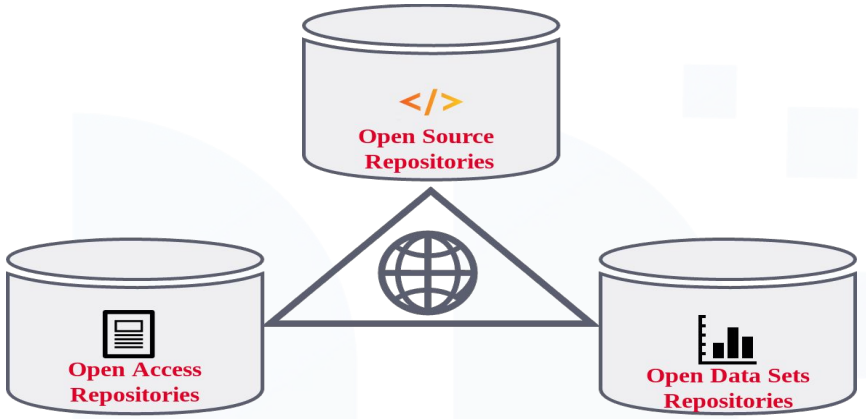
➔ created

- **during the research process**
- **for a research purpose**

Software in research

➔ used for research

FAIR4RS output: Gruenpeter et al. Defining Research Software: a controversial discussion (Version 1). Zenodo. https://doi.org/10.5281/zenodo.5504016



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

Software has multiple facets:

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

Software Heritage

Goal: Make software a first class output

French national plan for Open Science

ADORE .SOFTWARE

Prix science ouverte du logiciel libre de la recherche
Récompense dix logiciels d'équipes françaises

<ReSA> Research Software Alliance

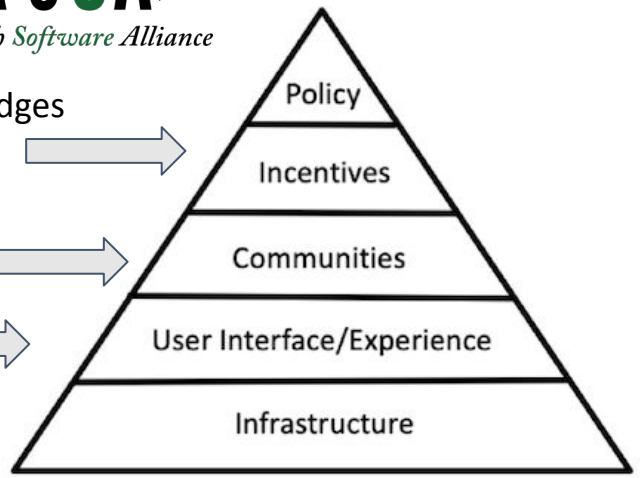
Awards & Badges

Credit - FORCE11 WG

RDA / FAIR4RS / FAIR-IMPACT

CodeMeta / CFF

SIRS report / FAIRCORE4EOSC



Pyramid from Strategy for Culture Change: Brian Nosek (2019)
https://www.cos.io/blog/strategy-for-culture-change

Inria

RSE

Software Sustainability Institute

CERN

CSC

swMATH an information service for mathematical software

DataCite

SCHLOSS DAGSTUHL Leibniz-Zentrum für Informatik

Software Heritage

eosc | FAIRCORE4EOSC

grnet

Data Archiving and Networked Services DANS

OpenAIRE

Why are we here? A plurality of needs

Researchers

- **archive and reference** software used and created in articles
- **find** useful software
- **get credit** for developed software
- **verify/reproduce/improve** results

Laboratories/teams

- **track** software contributions
- **produce** reports
- **maintain** web page

Research Organization

know its **software assets** for:

- technology transfer,
- impact metrics,
- Strategy

Curators

- **verify** and **curate** software metadata
- **provide** documentation on software curation
- **monitor** research teams' production

Software Heritage

What is at stake?

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)

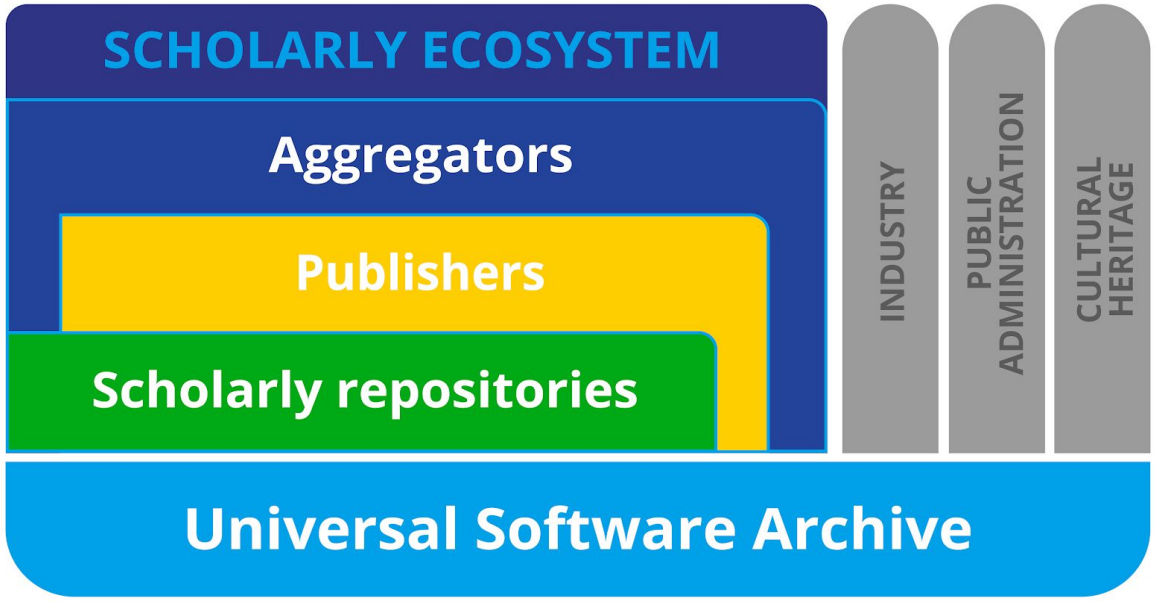
Software Heritage

Why archiving (research) software is important?

Software Heritage

Source code is fragile

SIRS report: European Commission, Directorate-General for Research and Innovation, *Scholarly infrastructures for research software : report from the EOSC Executive Board Working Group (WG) Architecture Task Force (TF) SIRS*, Publications Office, 2020, **https://data.europa.eu/doi/10.2777/28598**
Video: EOSC Software Infrastructures for Research Software: J. B. Gonzalez Lopez (CERN)



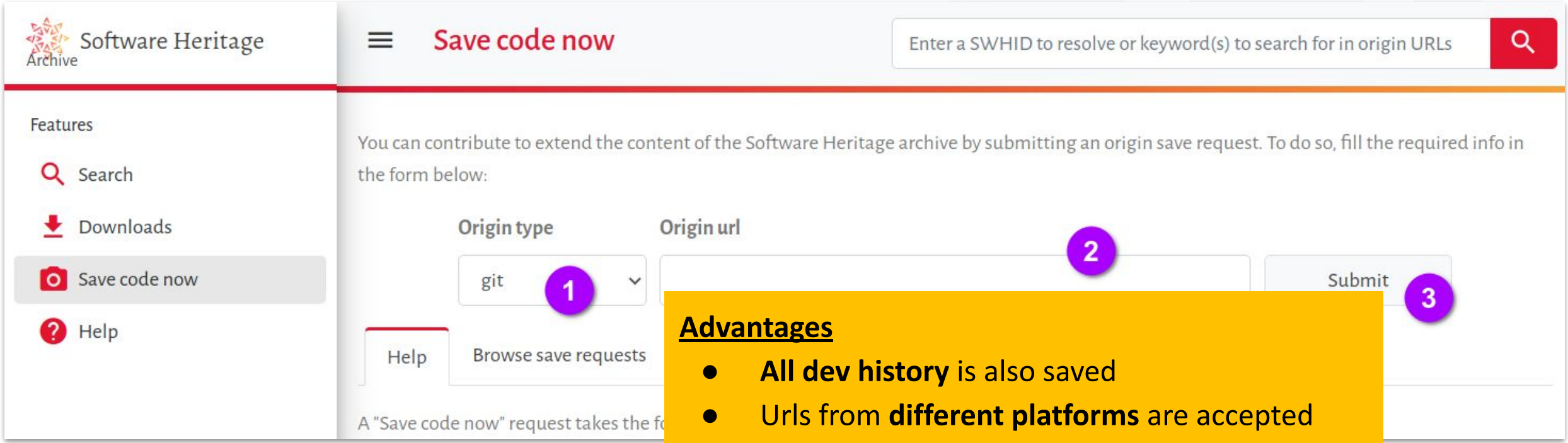
Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

archive.softwareheritage.org

Collect, **preserve** and **share** all software source code
Preserving our heritage, enabling better software and better research for all

Saving ~~your~~ any code now!

https://archive.softwareheritage.org/save/

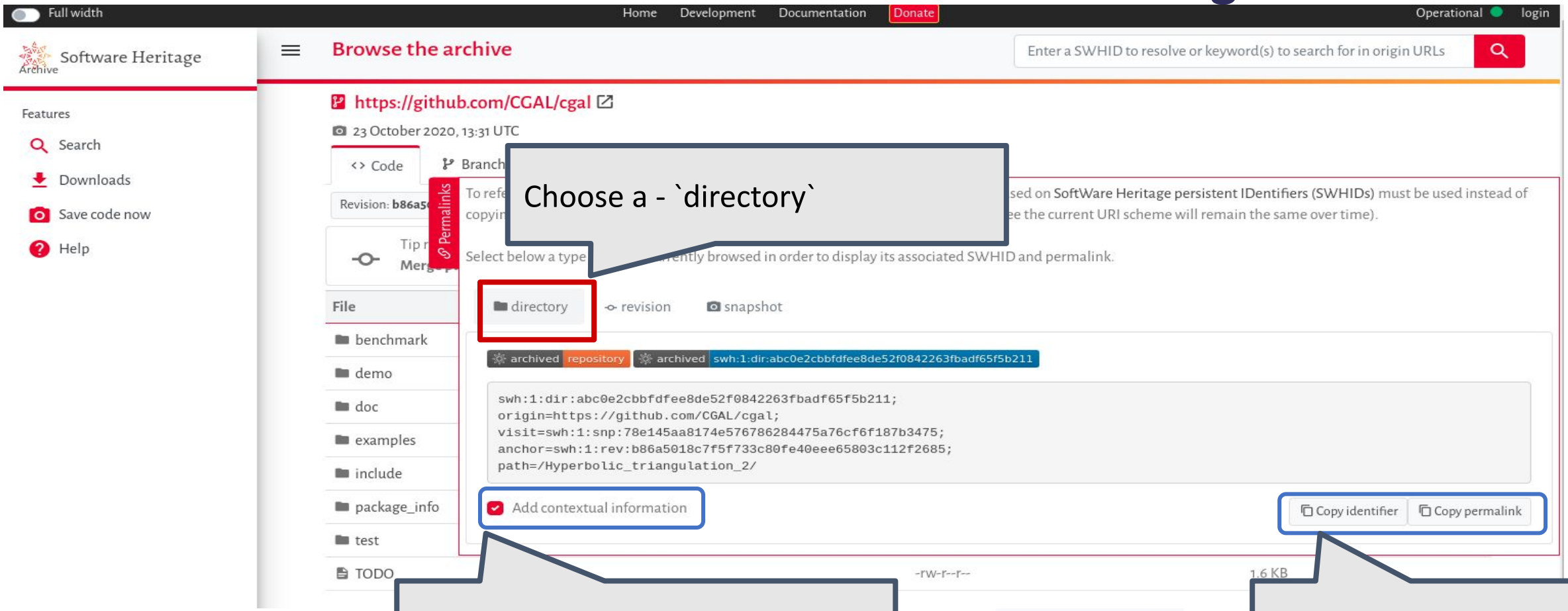


Advantages

- **All dev history** is also saved
- Urls from **different platforms** are accepted
- PID to **reference** specific pieces of code (even algorithms)

Software Heritage

Choose a SWHID on Software Heritage



Software Heritage

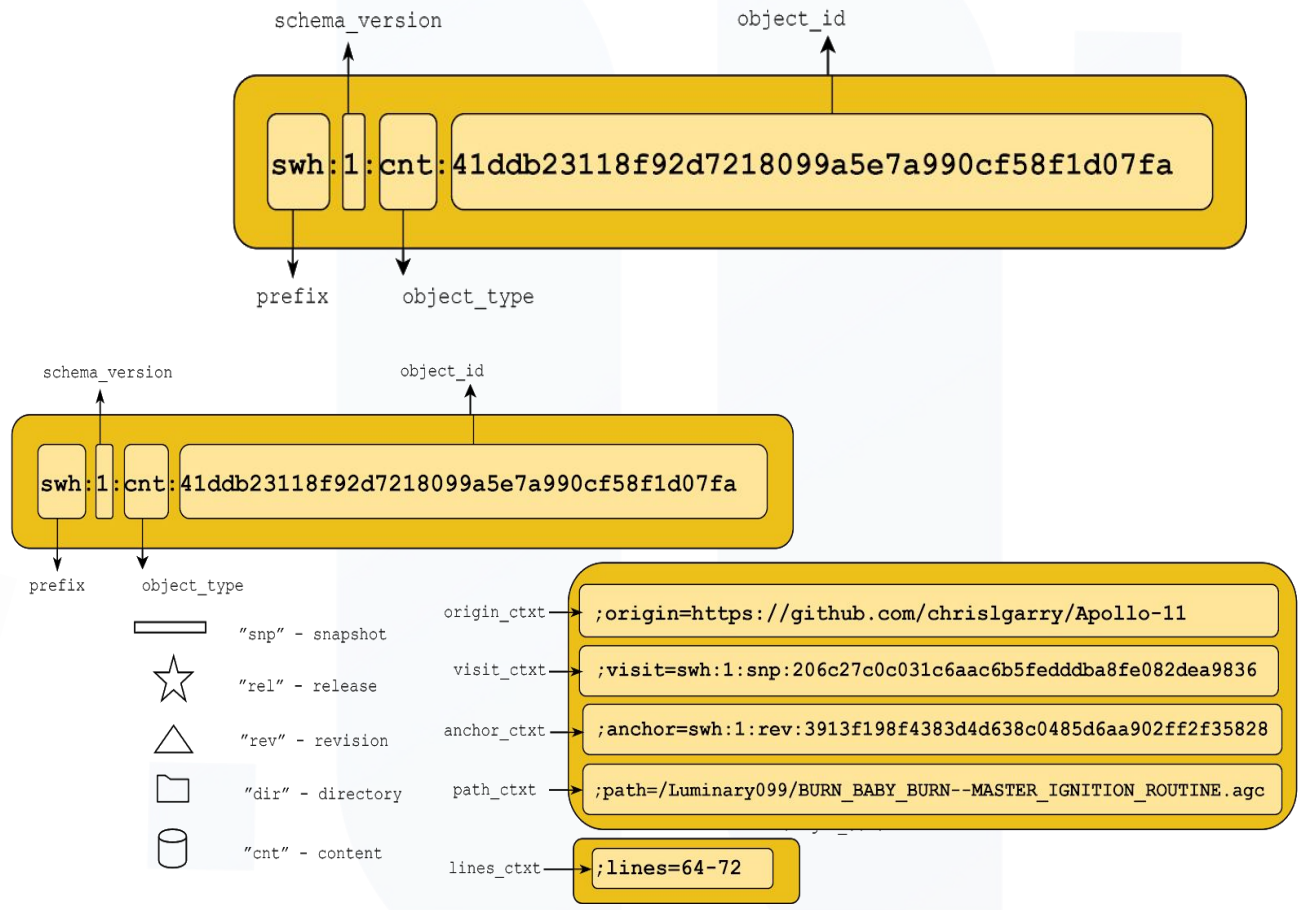
The SoftWare Heritage/Hash ID - a.k.a SWHID

SWHIDs are **intrinsic identifiers** which are intimately bound to the designated object, they do not need a register, only agreement on a standard.

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

[Intrinsic vs. extrinsic blog post]

Go to API endpoint



Intrinsic vs. extrinsic identifiers

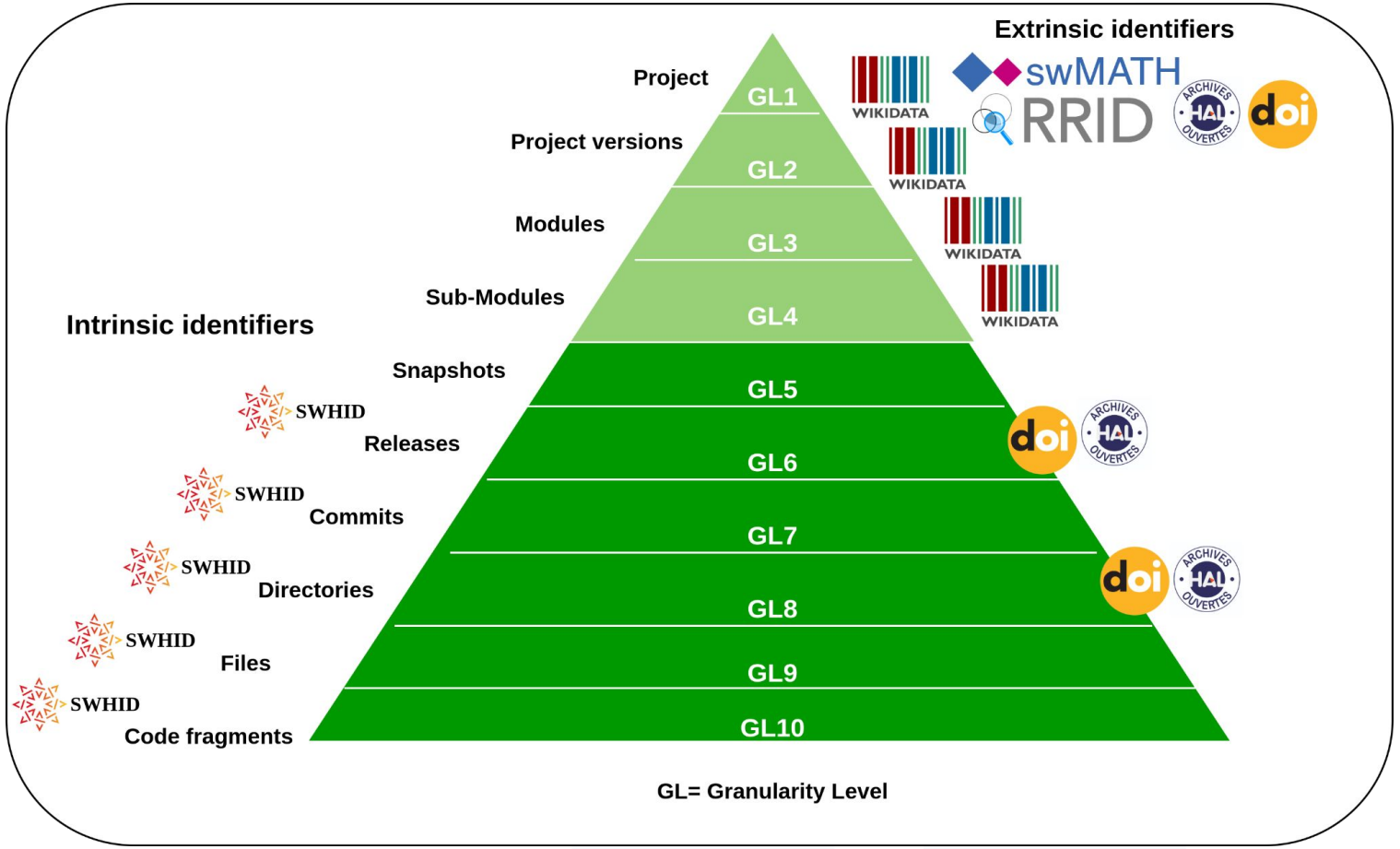
Software concept / project / collection

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

- Project versions (for example Python2 and Python3)
- Modules
- Sub-modules

Software artifact

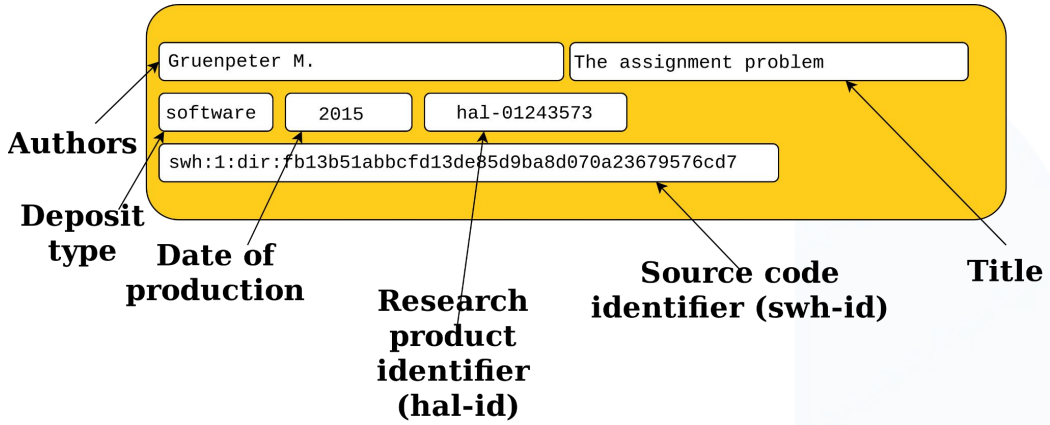
- Executable (download link)
- Software source code
  - Dynamic artifact - current development code
  - Archived copy
    - Snapshot (all branches, all dev history)
    - Release / Package
    - Commit- a specific point in development history
    - Directory
    - File
    - Algorithm



Research Data Alliance/FORCE11 Software Source Code Identification WG et al. (2020). Use cases and identifier schemes for persistent software source code identification (V1.1). *Research Data Alliance*. https://doi.org/10.15497/RDA00053

Software Heritage

For which use case: reference vs. citation



Archive & Index

- metadata record (extrinsic)
- artifact itself (intrinsic)

Credit & Attribution

- a metadata record
- all authors & contributors

Reuse & Reproducibility

- a specific artifact
- appropriate metadata/docs

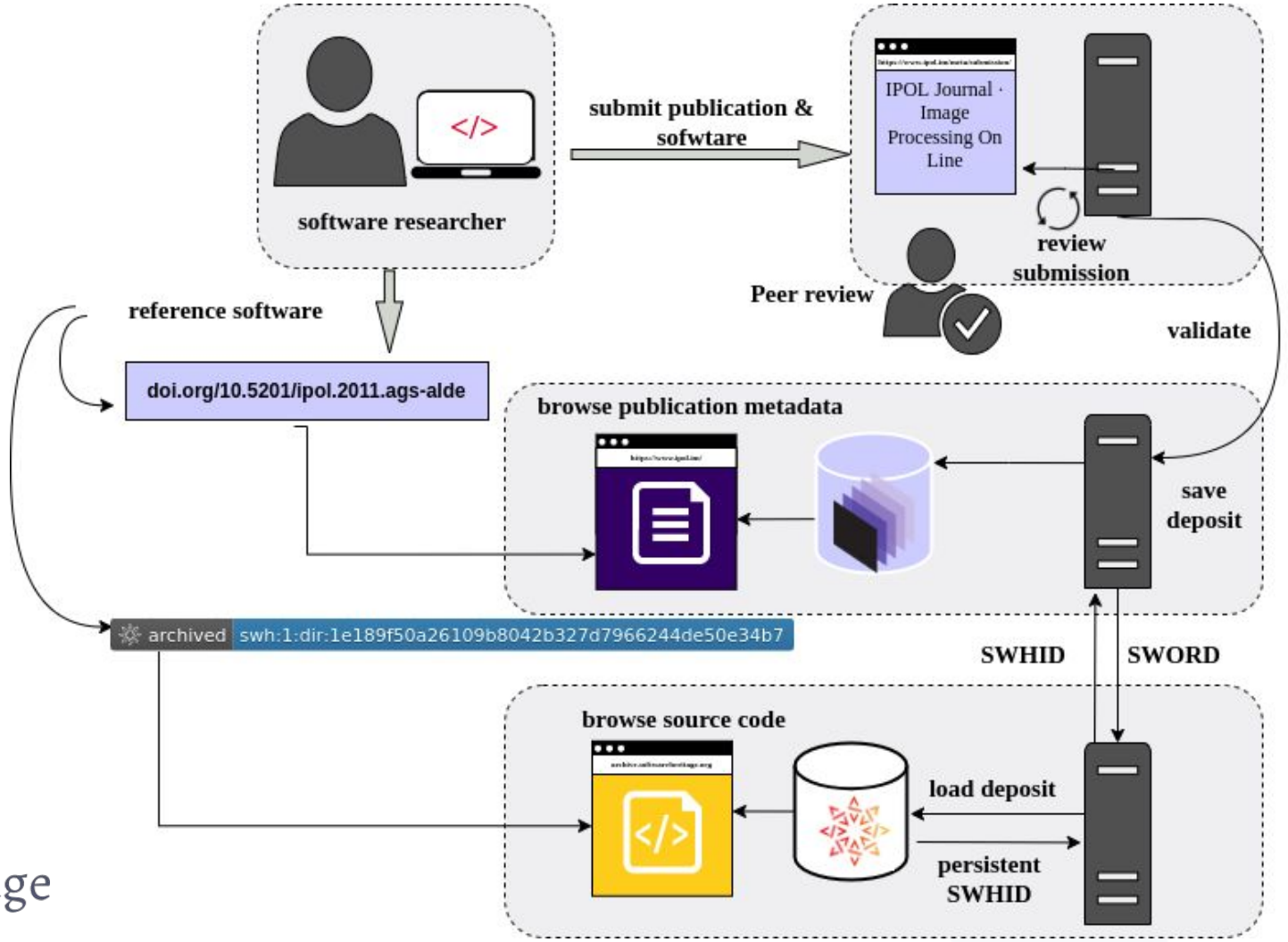
DOI and SWHID on IPOL articles

- Journal Image Processing On Line (IPOL, https://www.ipol.im/)

- Research software packages are identified with:
  - The article DOI: (https://doi.org/10.5201/ipol.2021.286)
  - The software SWHID: The publisher deposits the software in Software Heritage with the DOI as an origin (https://archive.softwareheritage.org/swh:1:dir:2cb75d8c95eb61d047d89428d0ec40a2286c0311;origin=https://doi.org/10.5201/ipol.2021.286;visit=swh:1:snp:23a5f7ee209b593e9b3e60ebe2bc42f1e6b76ff3;anchor=swh:1:rel:2de235c8fc3dd527cfaaba5cbf1d8144fee14f40)

- Links from the paper and metadata DOI to:
  - the software deposit and its SWHID,
  - the live demo of the software (in the demo tab)

Publish software workflow: two PIDs with one stone



Citation with the biblatex-software package

BibLaTex style extension for software

Youtube tutorial

[Software Release] B. Langmead and S. L. Salzberg, *Bowtie2* version 2.4.2, Oct. 2022. LIC: GPL. URL: http://bowtie-bio.sourceforge.net/bowtie2/index.shtml, VCS: https://github.com/BenLangmead/bowtie2, SWHID: ⟨swh:1:rel:97bacffeaa6e7c3f574ce5b566daba82aa18a11f;origin=https://github.com/BenLangmead/bowtie2;visit=swh:1:snp:c25778cfefc086c63c6f78eed230d0b9c88876ee⟩.

[Software excerpt] MIT Instrumentation Laboratory, "AGC Luminary routine for changing LEM asset during landing", from *Apollo 11 Guidance Computer (AGC) source code for the command and lunar module* 1967. VirtualAGC project. LIC: Public Domain. URL: https://www.ibiblio.org/apollo, VCS: https://github.com/virtualagc/virtualagc, SWHID: ⟨swh:1:cnt:64582b78792cd6c2d67d35da5a11bb80886a6409;origin=https://github.com/virtualagc/virtualagc;anchor=swh:1:rev:007c2b95f301f9438b8b74d7993b7a3b9a66255b;lines=245-261⟩.

The Research Software MetaData guidelines #RSMD
is open to community review until **May 29th**
see https://rb.gy/u5e04

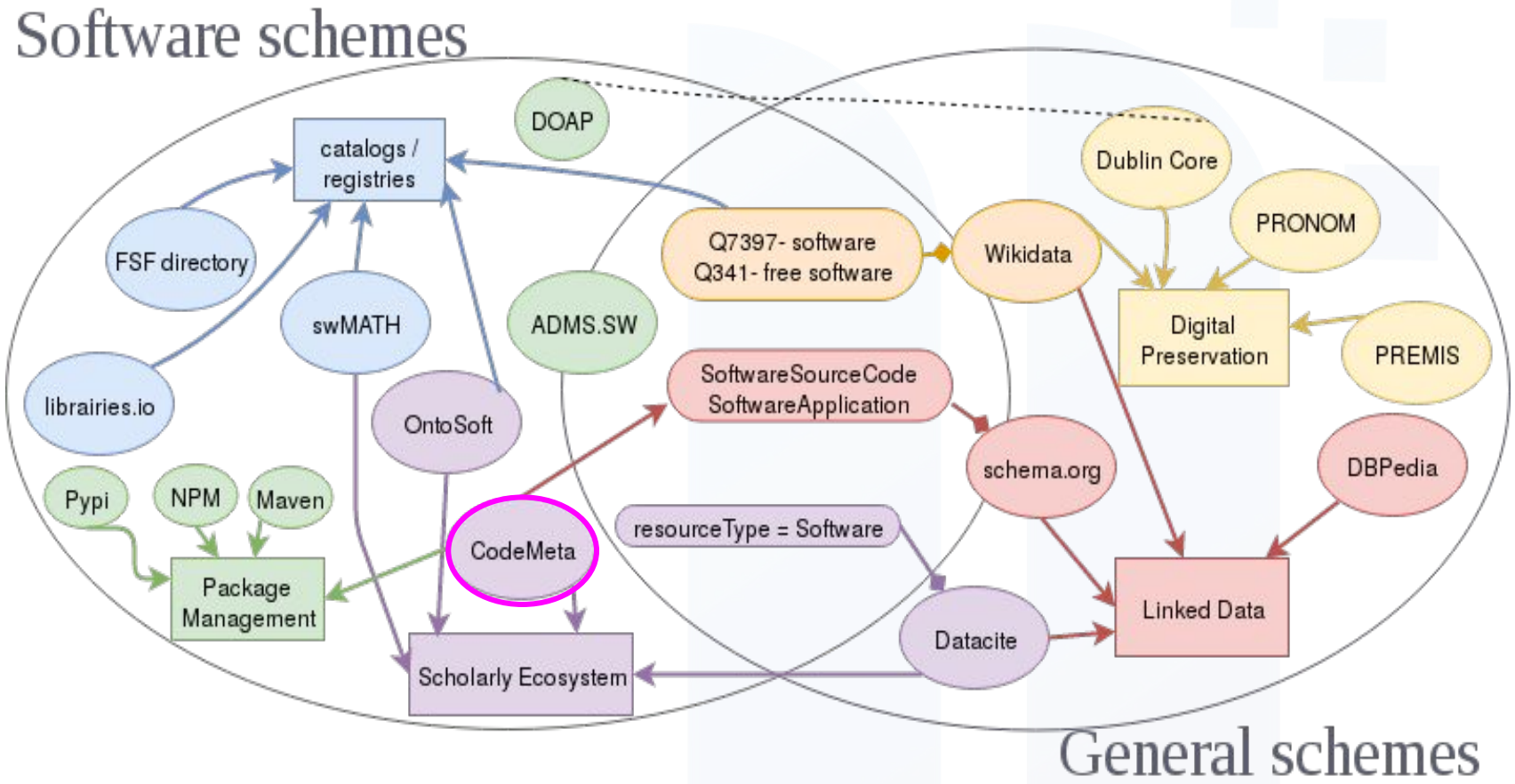
* Generated with wordcloud library in Pytho
using CodeMeta's crosswalk tabe

CodeMeta initiative - V1.0 released in 2017

- A subset of schema.org
- An academic community discussing software metadata
- A crosswalk table - mapping the metadata landscape

V2.0 released in 2020,

V3.0 is expected soon!!



Gruenpeter M. and Thornton K. (2018) Pathways for Discovery of Free Software (slide deck from LibrePlanet 2018). https://en.wikipedia.org/wiki/File:Pathways-discovery-free.pdf

Wrap up

- ★ **Archive** your code!
  - Save on SWH https://save.softwareheritage.org/ Tutorial How To
  - Deposit, publish or register in a **scholarly infrastructure**
- ★ **Adopt** good practices to *reference* & *cite* r#ResearchSoftware
  - SWHID with context
  - DOI + SWHID - version, release, file, lines biblatex-software Tutorial
- ★ **Join** the #ResearchSoftware community & review the #RSMD_guidelines
- ★ **Spread the word** and let's start recognizing software in academia

Software Heritage

Thank you!

morane@softwareheritage.org
@moraneottilia

Cite with: 10.5281/zenodo.7964043

FAIR-IMPACT
Expanding FAIR Solutions across EOSC

@twitter Linkedin

Funded by
the European Union