Reproducibility in Computer Science

Where we come from, where we are, where we go

Roberto Di Cosmo

Director, Software Heritage
Inria and Université Paris Cité

 @rdicosmo roberto@dicosmo.org

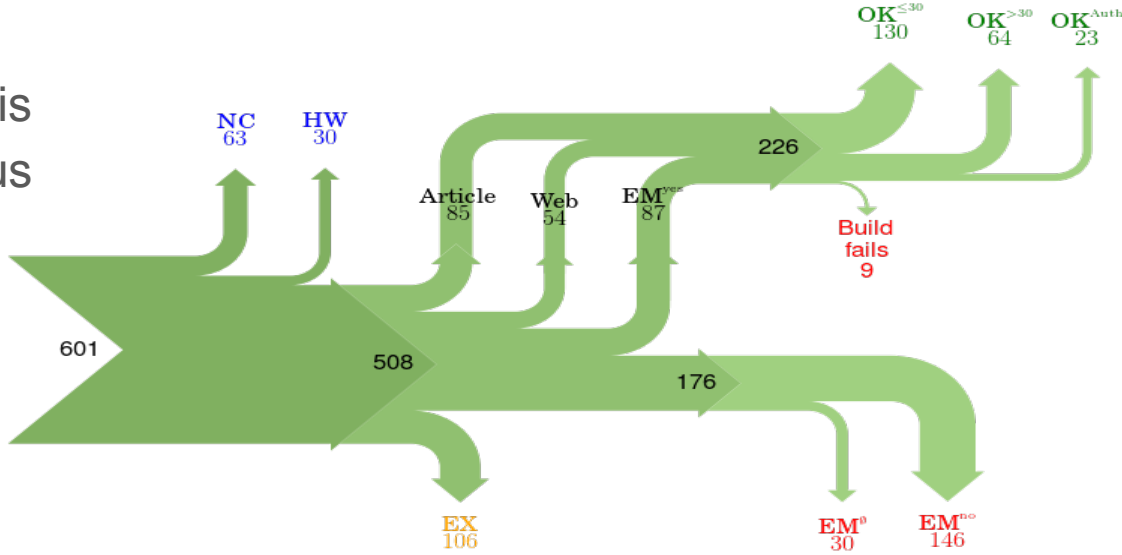
State of the art in the field ~2010

Software Engineering

2009: Carlo Ghezzi, 60% of ACM TOSEM papers have code, only 20% installable

Computer systems research

2014: Christian Collberg, analysis of **~600 papers** in prestigious venues, **~200** ***cannot even find the source code!***



Awareness and actions

Artifact Evaluation Committees

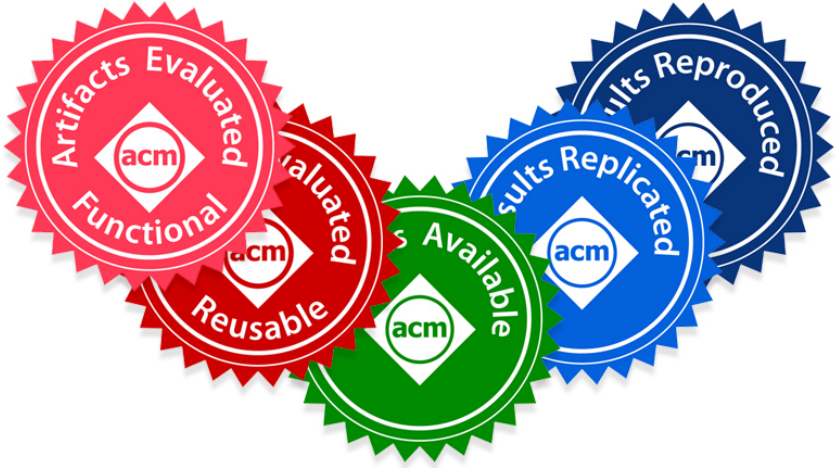
2011: run the first time as an award at ESEC-FSE (J. Vouillon and R. Di Cosmo)

2012-today: the process generalizes to a list too long to maintain

ACM software badges for publications

See home page for details.

- Very good intentions, but …
- Perfectible implementation



A few key issues in reproducibility (*there are many more!*)

Archive

Ensure **long term availability** of artifacts **with the development history**

Reference

Ensure **precise identification** of artifacts at **various levels of granularity**

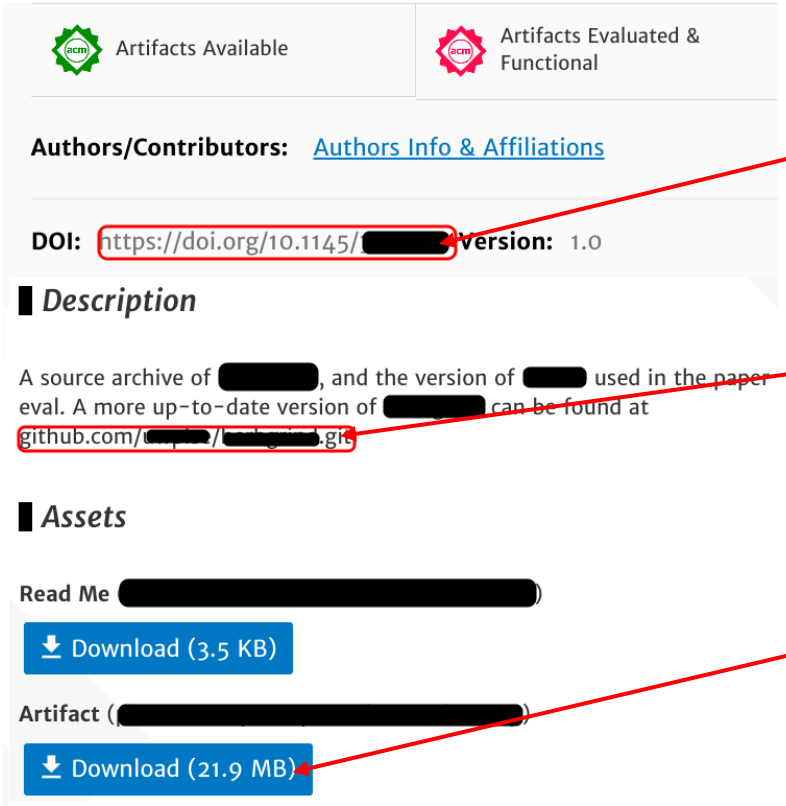
Describe

Provide **detailed description** (machine readable metadata)

and **proper documentation** (build instructions, dependencies, configuration)

and also ***link to relevant papers***

Not there yet, event for these basic needs - ACM DL



Only a DOI identifier …

… does not fill the **author's needs**

Zip file with source code, **looses the version control history!**

Not there yet, event for these basic needs: Papers with code

The Forward-Forward Algorithm: Some Preliminary Investigations

nebuly-ai/nebullvm • PyTorch • NA 2022

The aim of this paper is to introduce a new learning procedure for neural networks and to demonstrate that it works well enough on a few small problems to be worth further investigation.

★ 4,944

8.64 stars / hour

Paper

Code

Not an archive!

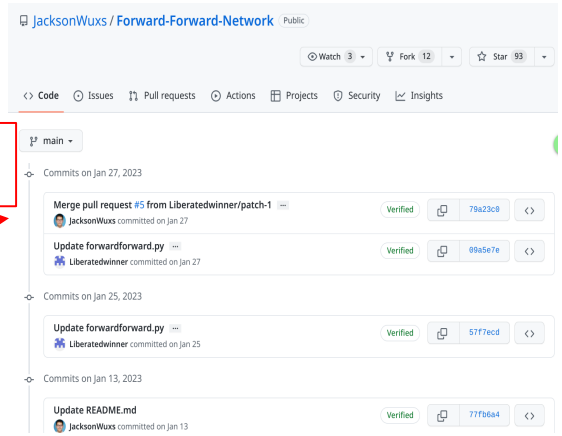
Code

Edit

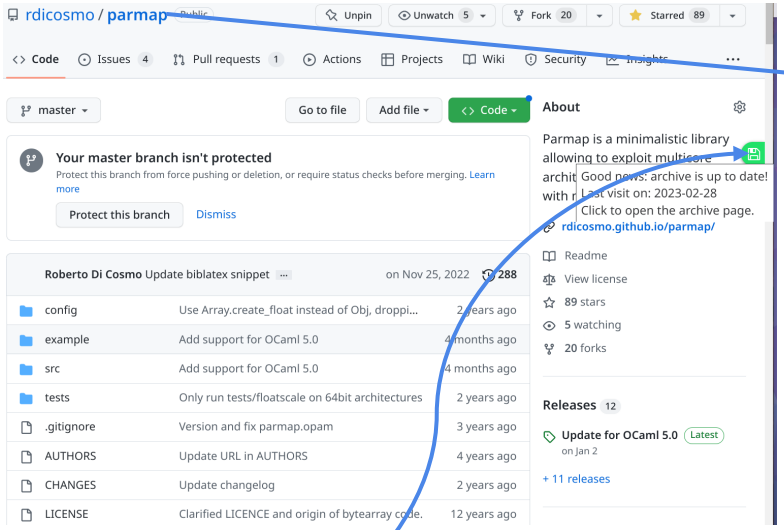
nebuly-ai/nebullvm	★ 4,944	PyTorch
keras-team/keras-io	★ 2,128	TensorFlow
mohammadpz/pytorch_forward_forward	★ 1,190	PyTorch
JacksonWuxs/Forward-Forward-Network	★ 93	PyTorch
EscVM/EscVM_YT	★ 48	PyTorch

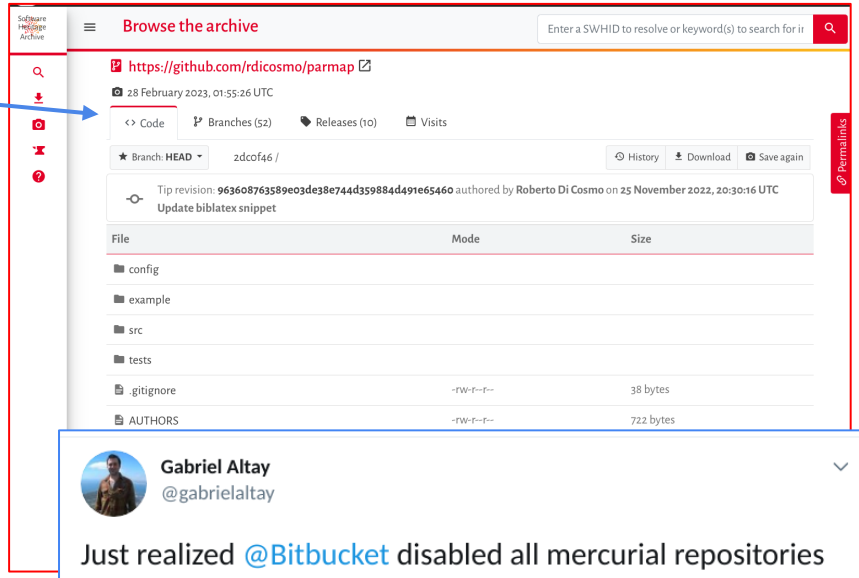
See all 6 implementations

Which version?



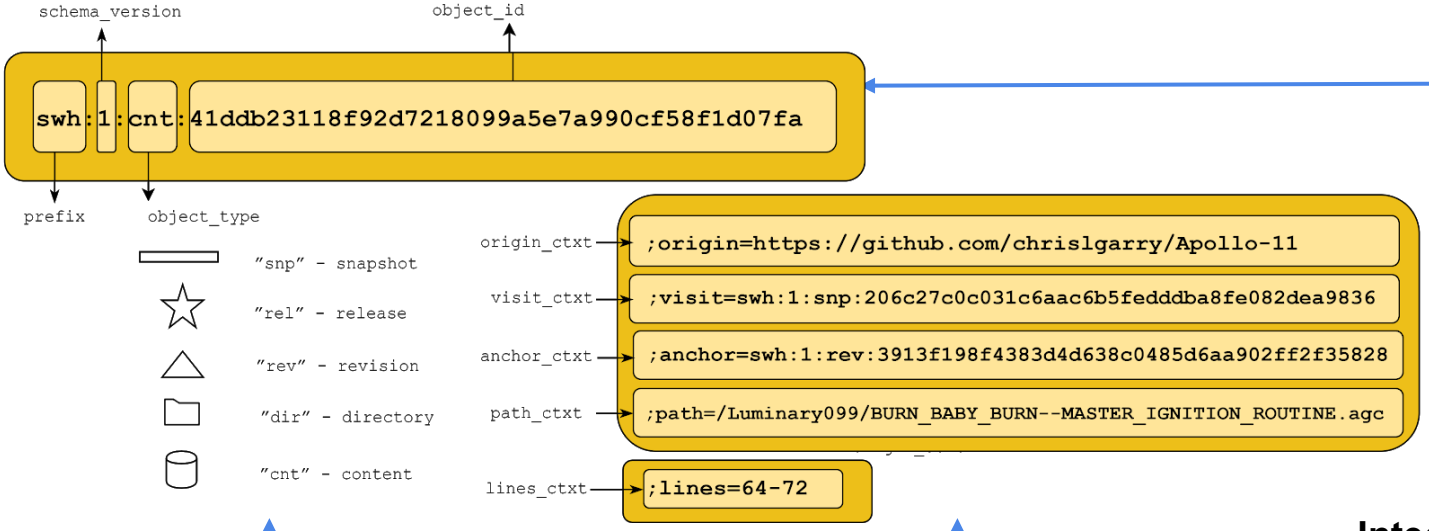
We can and must do better: **archive** in Software Heritage





- Regular crawling
- **One click** archival via **updateswh** browser extension
- Webhooks for BitBucket, Gitea, GitHub, GitLab, Sourceforge

We can and must do better: **reference** in Software Heritage



SWHID

Cryptographic, **intrinsic**

Soft**W**are
Hash
IDentifier

All levels of **granularity**

Rich **contextual** (optional) **information**

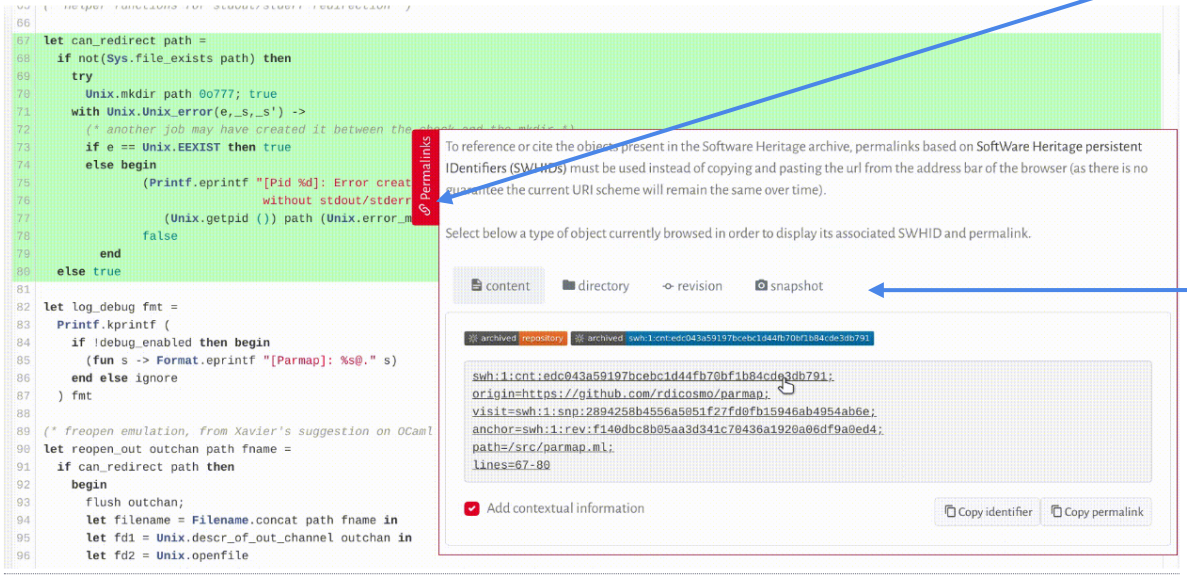
Integrity of identified artifact can be verified independently

Standard at https://swhid.org

We can and must do better: **reference** in Software Heritage

Getting the SWHID for a code fragment

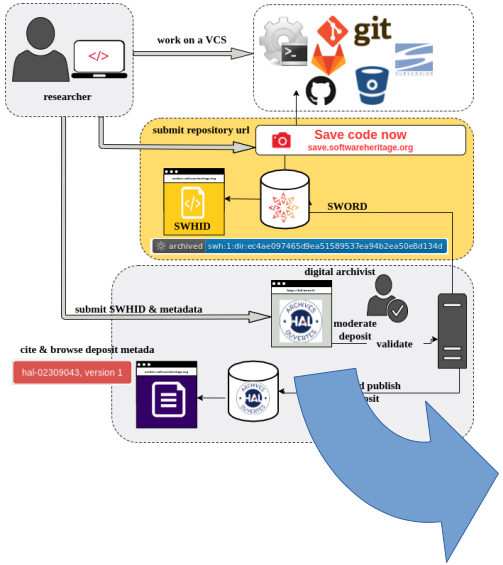
You can also get the SWHID of a file, or a code fragment inside a file. For this, navigate first to the file, select (optionally) the code fragment of interest by clicking on the line number of the first line, and shift-clicking on the line number of the last line. Then, pull out the red Permalinks tab and copy the SWHID identifier or the corresponding permalink.



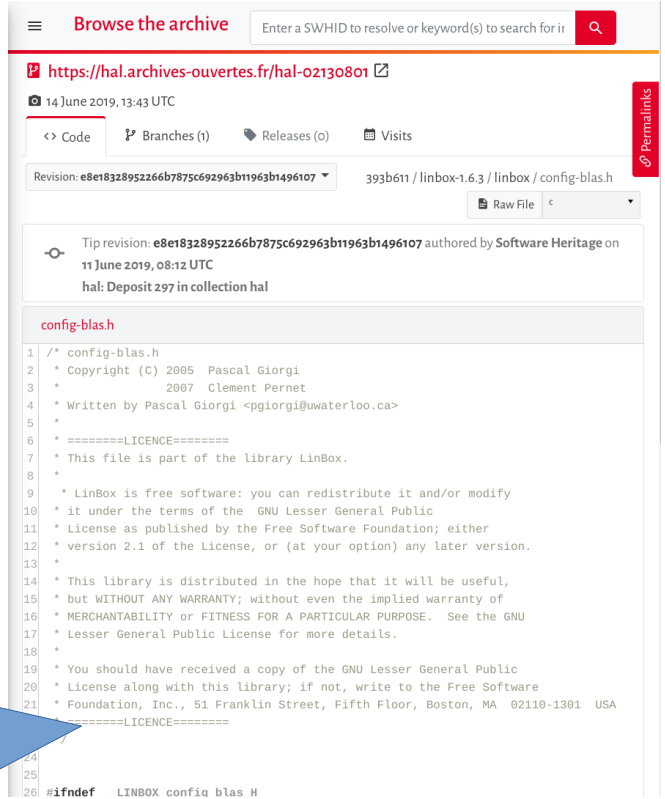
~ 30 billion SWHIDs can be found in Software Heritage

All levels of **granularity**:
- repository snapshot
- release
- revision
- directory
- file content
- code fragment

In France : HAL + Software Heritage for describe and cite



https://hal.archives-ouvertes.fr/ha



swh:1:dir:393b611a1424f032e83569bf6762
371cfcf65

The way ahead

Archival and reference for **source code**

- **Technical barriers** are mostly solved issues (*over 6 years of work*)
- **Social barriers** still stand in the way (adoption, training, cost mutualization, …)

Thank you

Learn more

- Software Heritage: https://softwareheritage.org and the 2022 annual report
- HOWTO archive, reference, describe and cite research software: https://bit.ly/swh-howto-research
- Software deposit and metadata curation: HAL-SWH Webinar, July 2022
- Deuxième plan national pour la Science Ouverte: official website
- Software Pillar session in OSEC 2022: official website
- EOSC SIRS report: https://data.europa.eu/doi/10.2777/28598
- Roberto Di Cosmo and Marco Danelutto. [Rp] Reproducing and replicating the OCamlP3l experiment. ReScience C, 6(1):#2, April 2020. [link]