

Beyond FAIR Data: Making Research Software a First-Class Research Output

Roberto Di Cosmo

Director, Software Heritage
Inria and Université Paris Cité

24 July 2026



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

- 1 Introduction
- 2 (Bad) SOTA
- 3 Assessing the needs and a strategy to address them
- 4 Meet Software Heritage: universal archive of source code
- 5 Demo time!
- 6 Practical tools for researchers
- 7 Shedding some light on the reference/identifiers confusion
- 8 Actions

Short Bio: Roberto Di Cosmo

Computer Science professor in Paris, now working at INRIA

- 35+ years of research (Theor. CS, Programming, Software Engineering, Erdos #: 3)
- 25+ years of Free and Open Source Software
- 15+ years building and directing structures for the common good



1999 *DemoLinux* – first live GNU/Linux distro

2007 *Free Software Thematic Group*

150 members 40 projects 200Me

2008 *Mancoosi project* www.mancoosi.org

2010 *IRILL* www.irill.org

2015 *Software Heritage* at INRIA

2018 *National Committee for Open Science*, France

2021 *EOSC Task Force on Infrastructures for Software*,
European Union

COMMENT | 07 October 2025

Stop treating code like an afterthought: record, share and value it

Scientists, research institutions, funders, libraries and publishers must all improve software practices.

By [Roberto Di Cosmo](#), [Sabrina Granger](#), [Konrad Hinszen](#), [Nicolas Iullien](#), [Daniel Le Berre](#), [Violaine Louvet](#), [Camille Maumer](#), [Clémentine Maurice](#), [Raphaël Monat](#) & [Nicolas P. Rougier](#) 



Illustration: Phil Wheeler

A *Nature* call to action

Nature commentary, October 2025

[doi:10.1038/d41586-025-03196-0](https://doi.org/10.1038/d41586-025-03196-0)

This talk is the *how-to* behind that call: how to **archive**, **reference**, **describe**, **cite** and **credit** research software — concretely, and today.

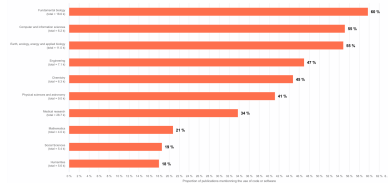
Software is a pillar of Open Science

Software powers modern research

Proportion of publications in France that mention the use of code or software by discipline

Sort by:

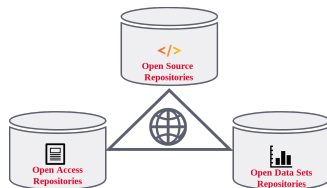
☐ Highest volume ☒ Highest use rate



Over 20% of articles using software across all disciplines share it

2024 French Open Science Monitor

Key pillar: software



Links are **important**

Nota Bene

software may be a *tool*, a *research outcome* and a *research object*

access to the *source code* is essential!

Preserving (the history of) source code is necessary for *reproducibility*

Software Source Code is Precious Knowledge

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

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

Apollo 11 source code (excerpt)

```
P63SP0T3      CA      BIT6      # IS THE LR ANTENNA IN POSITION 1 YET
               EXTEND
               RAND      CHAN33
               EXTEND
               BZF      P63SP0T4      # BRANCH IF ANTENNA ALREADY IN POSITION 1

               CAF      CODE500      # ASTRONAUT: PLEASE CRANK THE
               TC      BANKCALL      # SILLY THING AROUND
               CADR      GOPERF1
               TCF      GOTOP00H      # TERMINATE
               TCF      P63SP0T3      # PROCEED SEE IF HE'S LYING

P63SP0T4      TC      BANKCALL      # ENTER INITIALIZE LANDING RADAR
               CADR      SETPOS1

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

Quake III source code (excerpt)

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this
    // can be removed

    return y;
}
```

Len Shustek, Computer History Museum

2006

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

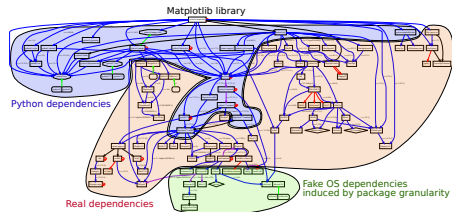
Software source code is *not* data (and FAIR is not the answer)

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
 - *research software* a thin top layer
- sophisticated *developer communities*



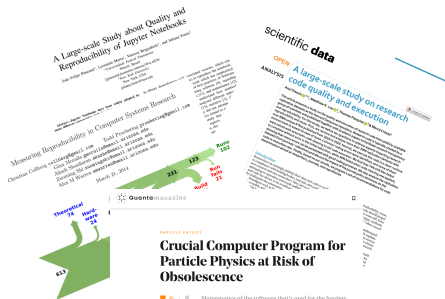
The human side

design, algorithm, code, test, documentation, community, funding
and so many more facets ...

- 1 Introduction
- 2 (Bad) SOTA
- 3 Assessing the needs and a strategy to address them
- 4 Meet Software Heritage: universal archive of source code
- 5 Demo time!
- 6 Practical tools for researchers
- 7 Shedding some light on the reference/identifiers confusion
- 8 Actions

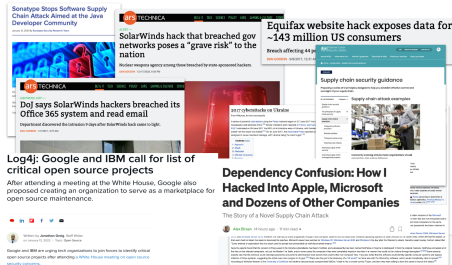
How are we managing our software ?

Reproducibility, maintenance in Academia



(articles: [here](#), [here](#), [here](#) and [here](#))

Security, integrity, traceability in Industry



Can they track the software that they

- ship, use, acquire
- has that bug or vulnerability

Pressure to make research code available is now raising

Evaluation of software artefacts

from optional contest to standard practice



- Started as contest (ESEC/FSE 2011, winner *Vouillon and Di Cosmo*)
- Now a near-universal standard: [ACM Artifact Review and Badging v1.1](#) (5 badges, *left*)

Adopted across computer science communities:

- **PL/SE:** POPL, PLDI, OOPSLA, ICFP, ECOOP, ICSE, FSE, MSR, ISSTA, CAV
- **Systems:** OSDI, SOSP, NSDI, EuroSys, ASPLOS, MICRO, ISCA, HPCA
- **Security:** USENIX Security (availability mandatory in 2025), CCS, NDSS, IEEE S/P
- **HPC and DB:** SC ([Repro. Initiative](#)), SIGMOD ([ARI](#)), VLDB

Best-of-breed exemplars:

- **IPOL:** peer-reviewed papers ship code + runnable in-browser demo ([e.g. non-local-means denoising](#))
- **Graphics Replicability Stamp:** 478 graphics papers independently reproduced

EuroSys 2025: 93% of papers earned "Functional"; ~47% pursued "Results Reproduced"

- 1 Introduction
- 2 (Bad) SOTA
- 3 Assessing the needs and a strategy to address them
- 4 Meet Software Heritage: universal archive of source code
- 5 Demo time!
- 6 Practical tools for researchers
- 7 Shedding some light on the reference/identifiers confusion
- 8 Actions

Source code in Open Science: a plurality of needs to address

Researchers

- archive and reference software used 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 Organizations and/or Funders

know its software assets

- technology transfer
- funding strategy
- impact metrics
- career evaluation

Archive

Research software artifacts must be properly **archived**
make sure we can *retrieve* them (*reproducibility*)

Reference

Research software artifacts must be properly **referenced**
make sure we can *identify* them (*reproducibility*)

Describe

Research software artifacts must be properly **described**
make it easy to *discover* and *reuse* them (*visibility*)

Cite/Credit

Research software artifacts must be properly **cited** (*not the same as referenced!*)
to give *credit* to authors (*evaluation!*)

Archive and reference: some popular approaches that do not fit the bill

A - Since the 1970's 1990's

.zip or .tar file on:

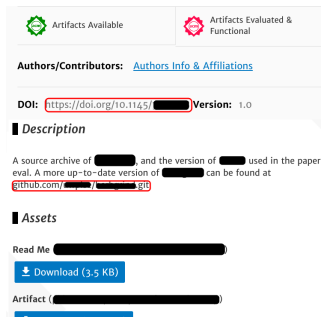
- ftp-server (e.g. [gnu](#))
- web page ([example](#))
- document archive (+ DOI [sample](#))

B - Since the 2000's

Rely on *software forges*

- institutional/project (e.g. [example](#))
- free commercial ones: BitBucket, GitHub, GitLab, ... (e.g. [parmap](#))

C: a mix of the two



Can get no satisfaction...

A *Poor user experience*

B *No preservation guarantee*

C *Can do so much better*

Forges are *not* archives!

2015: the first big bad news

Google Code and Gitorious.org shutdown: ~1M endangered repositories

- broken links in the web of knowledge (my papers too)

Big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250.000+ repositories (including research software)
- summer 2022: GitLab.com considers erasing **all** projects that are **inactive for a year**

In Academia too!

- 2021: Inria's old gforge is unplugged... **breaks the Opam build chain** for OCaml

We need a universal archive of software source code: now we have one!

- 1 Introduction
- 2 (Bad) SOTA
- 3 Assessing the needs and a strategy to address them
- 4 Meet Software Heritage: universal archive of source code**
- 5 Demo time!
- 6 Practical tools for researchers
- 7 Shedding some light on the reference/identifiers confusion
- 8 Actions



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Collect, preserve and share *all* software source code

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

Reference catalog



find and reference all
software source code

Universal archive



preserve and share all
software source code

Research infrastructure



enable analysis of all
software source code

Sharing the vision



And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Inria

Diamond sponsors



Platinum sponsors



Gold sponsors



Silver sponsors



Bronze sponsors



The largest software archive, a shared infrastructure

Cultural Heritage



Industry



Research



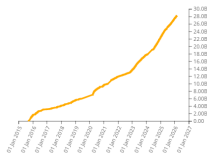
Public Administration



Software Heritage

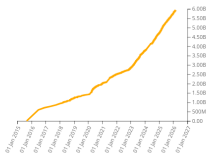
Source files

28,178,230,711



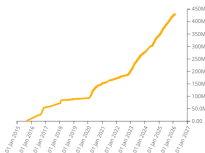
Commits

5,925,711,720



Projects

430,218,860



Directories

22,013,760,256

Authors

105,360,419

Releases

147,686,240

Bitbucket

1,925,997 origins

git

21,603 origins

R

21,113 origins

debian

128,719 origins

gn

5,947 origins

GitHub

137,564,899 origins

GitLab

3,982,586 origins

Guix

12,032 origins

GNU

354 origins

heptapod

1,068 origins

launchpad

329,908 origins

Maven

93,738 origins

NixOS

12,032 origins

npm

1,799,296 origins

Box

4,083 origins

Phabricator

192 origins

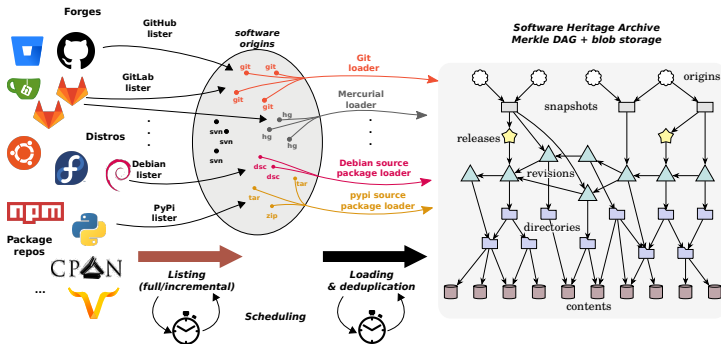
python

410,582 origins

SOURCEFORGE

308,990 origins

Address common Open Science and Open Source needs: archival

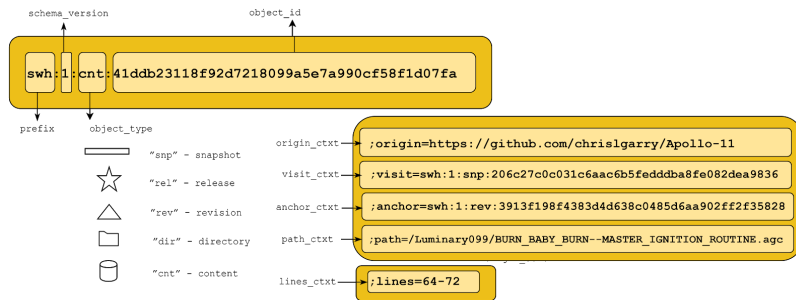


Global development history permanently archived in a uniform data model

- over 28 billion unique source files from over 430 million software projects
- ~3PB (compressed) blobs, ~60 B nodes, ~1 trillion edges

Meet the SWHID identifier

Software Hash Identifiers (SWHID)



[link to full docs](#)

25+B

intrinsic,
decentralised,
cryptographic

Full fledged *source code references* for traceability, integrity and reproducibility

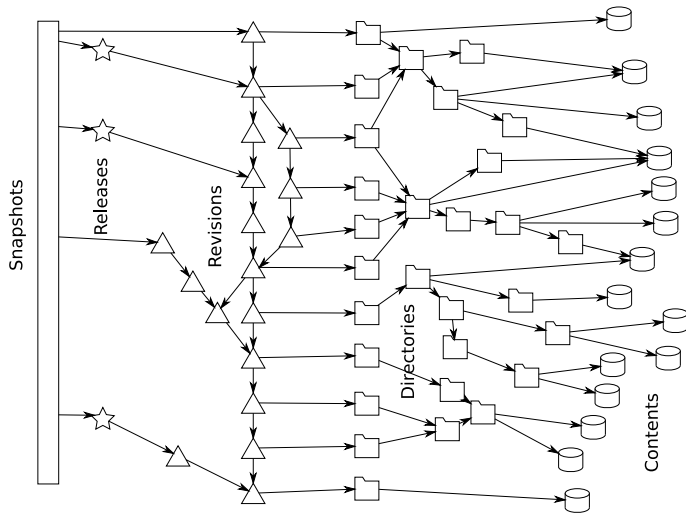
Examples: [Apollo 11 AGC excerpt](#), [Quake III rsqrt](#)
Guidelines available, see [the HOWTO](#)

- Linux Foundation [SPDX 2.2](#)
- IANA-registered "swh:"
- WikiData property [P6138](#)

Now an ISO standard: [ISO/IEC 18670:2025](#)
implementation: [swh.model](#)

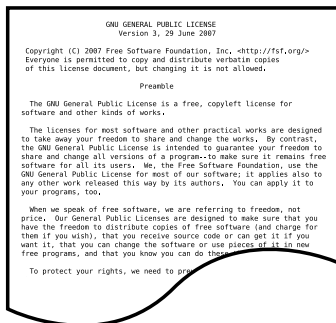
ref.

How a SWHID is built: a Merkle DAG of the development history



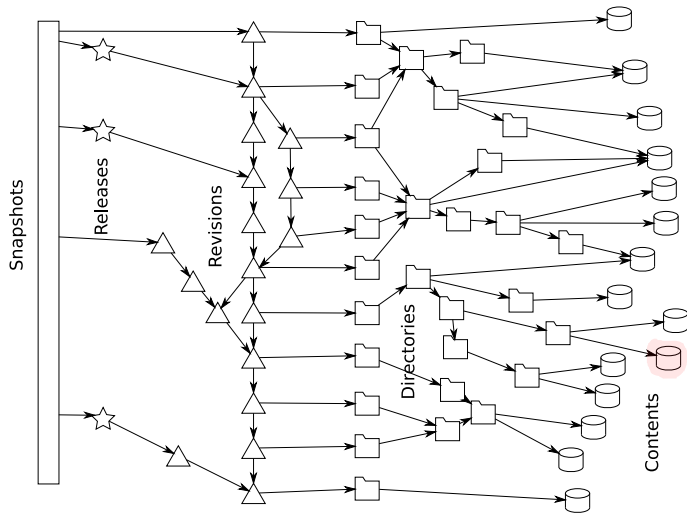
How a SWHID is built: a Merkle DAG of the development history

Contents



sha1: 8624bcdade55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: **94a9ed024d385...**
length: 35147

How a SWHID is built: a Merkle DAG of the development history



How a SWHID is built: a Merkle DAG of the development history

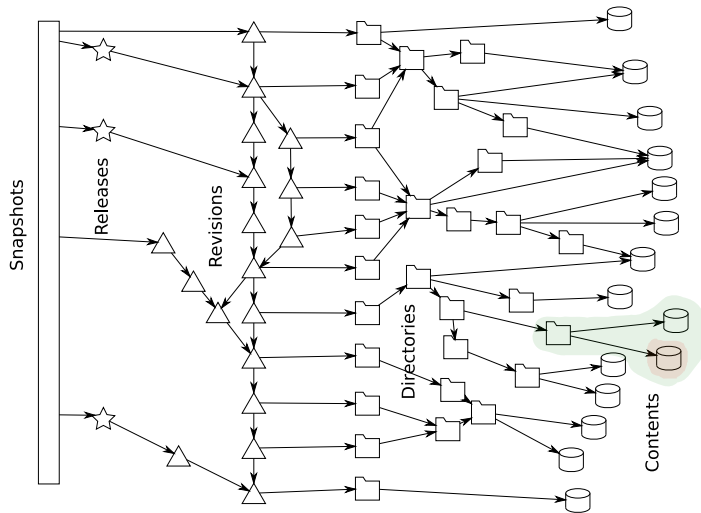


Directories

```
100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d0a34af6f52cf3cf6b0c2f7bd0648fbd255e77f AUTHORS
100644 blob 94a9ed024d3859793618152ea559a168bbcbb5e2 LICENSE
100644 blob d9b2665a435a43f8a79a84e0867751dfb095c7bb MANIFEST.in
100644 blob 524175c2bad0b35b975f79284c2f5a6d5eaf2eb4 Makefile
100644 blob 5c7e3a5bbddb038682ba7793f440492ed9678bb3 Makefile.local
100644 blob 8617980629cd24e6080404f09aa749b085b3e07b README.db_testing
100644 blob 76b29f94cf815e0869c414d38d78d7ce08ec514e README.dev
040000 tree e1e10ecef948af0b93adb0372afc89f12e92618a bin
040000 tree 83e56d0beaf7793c77a45a345c80fcb8af503013 debian
040000 tree a34c9c4ba213f0cedc67f9816348d27955577af5 docs
100644 blob f2a6d32c6135aa7287bbd76167b01df2ae4f1539 requirements.txt
100755 blob eee147c36caf1bbc2d820da8dc026cb5b68180bc setup.py
040000 tree 224bb4c1f4c67fca1d160bff2d06094e7e1abf3 sql
040000 tree 8631c9cd77bbe993168107ab5baf51f40c6300be swh
040000 tree 8fb905b56ba8ed692f1209b2773b474c6c1d66c1 utils
```

id: 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d

How a SWHID is built: a Merkle DAG of the development history



Revisions

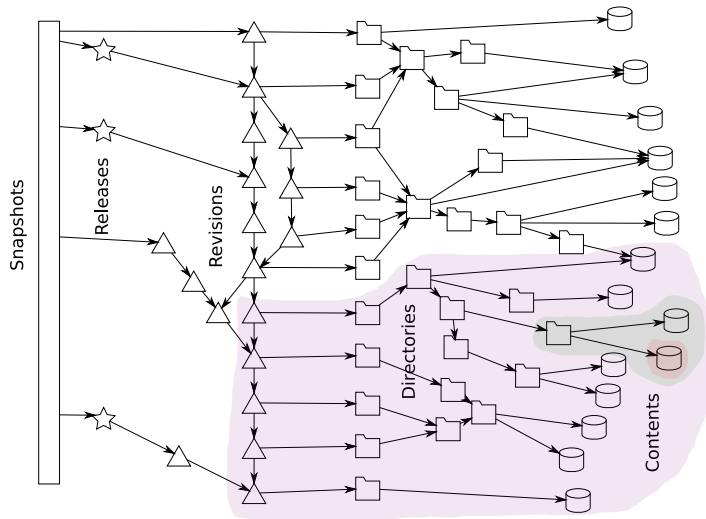
Details	Changes	Files
SHA: 963634dca6ba5dc37e3ee426ba091092c267f9f6		
Author: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Committer: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Subject: provenance.tasks: add the revision -> origin cache task		
Parent: fc3a8b59ca1df424d860f2c29ab07fee4dc35d10 : test_storage: properly pipeline origin and cont...		
provenance.tasks: add the revision -> origin cache task		
swf/storage/provenance/tasks.py  77		

tree 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d
parent fc3a8b59ca1df424d860f2c29ab07fee4dc35d10
author Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200
committer Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200

provenance.tasks: add the revision -> origin cache task

id: 963634dca6ba5dc37e3ee426ba091092c267f9f6

How a SWHID is built: a Merkle DAG of the development history



How a SWHID is built: a Merkle DAG of the development history

Releases

tag v0.0.51
Tagger: Nicolas Dandrimont <nicolas@dandrimont.eu>
Date: Wed Aug 24 14:36:03 2016 +0200

Release sw.h.storage v0.0.51

- Add new metadata column to origin_visit
- Update sw.h-add-directory script for updated API [...]

commit c0c9f16b1e134f593e7567570a1761b156e6eb1d

object c0c9f16b1e134f593e7567570a1761b156e6eb1d
type commit
tag v0.0.51
tagger Nicolas Dandrimont <nicolas@dandrimont.eu> 1472042163 +0200

Release sw.h.storage v0.0.51

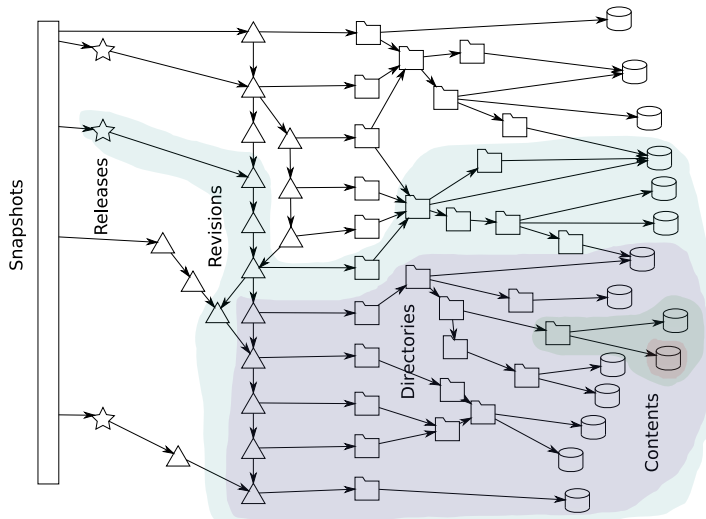
- Add new metadata column to origin_visit
- Update sw.h-add-directory script for updated API

-----BEGIN PGP SIGNATURE-----

iQIzBAABCAAdBQJXvZTNFhXuaWNvbGFzQGRhbmRyaW1vbnQuZXUACgkQ7AWLMo2+
neqorw//aq6SOb5DijEa+kWN3rXgVS+1K1vEVh1wNKAwx8eKJ7aX2kEiLDtt7uf
ahpZ6pz3q8nqs6aC1+YrxBfcih3L2YtrdZeWXWqr8xWNMaEoYDb8qaphwh8AD5t2
ICBIt2ujtXuCrDt93eKKPwvZxg+h80sMWy35Dr6jW7Z7K4Mu/PgGlyIHPY55yo
IGEndWno7Vfh1Vm6t1n5qB7l5mXRagA+becgdubTZ2xij+jpIUqC8cyqN3hm/fL
qsJ2mu8kyz3t8tG/H1/pV+I5OwBlNpO55TH0tuoJEvGPK/dHSP79QuHDZFkCao
kij6kAWyU80Mxb+nKVjLbrR3+yWBFj3Qp5a1/V8oOT6E1dALcNMpEaKCoKtMt
d/gMRax1I1/g0EDfnsW67G6sDwKPKPhgFVLQ3nV3GaQQTnu1RpMz006H9/tAwzC
Gg/K1PdHT4hz0i46wYPZyje0U2VXGFu6vU9vFQ4ZR/Wjn+0zMzdcRdrIJSUOMn
RpTTFUsbXUeXHGOpkgXhSYTnvp1gdPc76USTsK0aGe84AZm1Ik0mGrwXCVFpqlYo
nhhibB5HBNMoqyF6yTSOpUbyK70tpYRRUGKwDeRk0wKSxkWKUzGtKzy6jYqlJo29
gulwgZQif5qWQCBOontAL2+HvPfaVyckMejUhg62cP/+EHlvUk=
=kOxP
-----END PGP SIGNATURE-----

id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb

How a SWHID is built: a Merkle DAG of the development history



How a SWHID is built: a Merkle DAG of the development history

Snapshots

git show-refs

```
commit 08ffeb25770109525eb3ce21691466c53a1d9158 refs/heads/atime
commit ba543a24e3f9fe323a46c292cec4fcb61c67eb refs/heads/directory-listing-arrays
commit d69e0dbf892383ff6589b27fbelc05d27238d9c5 refs/heads/foo
commit c7ff9eea0eb22f8946908f5a8019f67de468e08 refs/heads/master
commit 7eca197fc66d2024047e54b1ed9e8b44361a0fc2 refs/heads/tmp-directory-add
commit 642a205f37de8505a85d427b53ee4fb2252e82e refs/heads/tmp/generic-releases
tag 20f043b1379cf768d966597799fd4907c757f755 refs/tags/v0.0.1
tag 72a21991a384e539996dbb867bfb0bee72aee2cd refs/tags/v0.0.10
tag 3590e0ca0ebb070e5b376705fa230bbfa4ffa5cc refs/tags/v0.0.11
tag 33378427a403ba569a67777b8d58f6674fbc6556 refs/tags/v0.0.12
tag 06f74652755b327cf590311c2bfa036cf3b4b35d refs/tags/v0.0.13
tag 5a6325fe86ab854b581d7442667d92a11e32f3bd refs/tags/v0.0.14
tag 586fba4e580b4f5fab05f599367643cbb1a9c7f refs/tags/v0.0.15
tag 8cd8b885f4098bf363177742bd289f660e5be51c refs/tags/v0.0.16
tag a542444ee3f0fbed35efb202fee035c809abc7d6 refs/tags/v0.0.17
tag 228a2f1650dd12222e556559462e1e06fc4993d9 refs/tags/v0.0.18
tag 606979a4ca05d497fc0d24aad0dce82636ef47c refs/tags/v0.0.19
tag 32bf5a59fc2a323baa6d5f15a6ad5382ec275a67 refs/tags/v0.0.2
tag 3147c3d31ec46cf6492f881e908b1237ebdff2c7 refs/tags/v0.0.20
tag 215ea50daba11e082e0b72e76eb4b6073a87908 refs/tags/v0.0.21
tag 3fb168c2072a5d6252124257a1e5dfc0f5ffa1df refs/tags/v0.0.22
tag 8cdbee8da4d73fc5d262789e460a16ac3c72aba4 refs/tags/v0.0.23
...
```

id: b464cad1b66fff266a37b46ea6e7a04b545e904b

Don't trust, verify: recompute the SWHID yourself

The SWHID is computed from the *content*: anyone can recompute it *offline* with the reference implementation (`swh.model`).

```
$ SWHID=swh:1:dir:HASH
# 1. download that directory as a tarball (the archive "vault")
$ curl -o tree.tar.gz "$ARCHIVE/api/1/vault/flat/$SWHID/raw/"
# 2. unpack it, then 3. recompute its SWHID locally
$ tar xf tree.tar.gz
$ swh identify --type directory HASH/
swh:1:dir:HASH      # identical: no need to trust the archive
```

- 1 Introduction
- 2 (Bad) SOTA
- 3 Assessing the needs and a strategy to address them
- 4 Meet Software Heritage: universal archive of source code
- 5 Demo time!**
- 6 Practical tools for researchers
- 7 Shedding some light on the reference/identifiers confusion
- 8 Actions

A walkthrough

- Browse (e.g. [Apollo 11 \[excerpt\]](#), your work [may be already there !](#))
- Trigger archival, use the [updateswh](#) browser extension, configure the webhooks
- Get and use SWHIDs ([full specification available online](#))
- Cite software with [biblatex-software](#) package from CTAN
 - [Overleaf ACMART template](#) available
- Example in journals: [article from IPOL](#)
- Example with Parmap: [devel on Github](#), [archive in SWH](#), [curated deposit in HAL](#)
- Extracting all the software products [for Inria](#), [for CNRS](#), [for CNES](#), [for LIRMM](#) or [for Rémi Gribonval](#) using [HalTools](#)
- [Curated deposit in SWH via HAL](#), see for example: [LinBox](#), [SLALOM](#), [Givaro](#), [NS2DDV](#), [SumGra](#), [Coq proof](#), ...
- Example use in research articles:
 - compare Fig. 1 and conclusions in [the 2012 version](#) and [the updated version](#)
 - SWHID in [a replication experiment](#)

IEEE TSE, submitted — 57

R. Di Cosmo roberto@dicosmo.org (CC-BY 4.0) Beyond FAIR Data 24/7/2026 20 / 31

A flagship life-sciences journal: code preservation built *into the publishing pipeline*, not left to author goodwill.

- *Before 2020*: code was *forked* into the GitHub org [elifesciences-publications](#) — no PID, manual licensing, did not work for non-GitHub repos
- *Since 2020*: every article's code is archived in Software Heritage; the SWHID is embedded in the JATS XML and rendered on the article landing page
- *Validation pipeline*: Schematron rules over the JATS XML *flag* every mention of code for vendors and *verify* that the SWHID matches the source repo before publication
- *SWHID structure example* (population genetics, [Zaidi et al.](#)):
=swh:1:dir:506ed3...;origin=github.com/Arslan-Zaidi/popstructure;anchor=swh:1:rev:1509a5... =
([browse the full qualified SWHID in the archive](#))
- *Researcher view* (P. Poulain et al., [eLife 90061](#), 2024): SWH hyperlinks in the [reviewed preprint](#) are transformed by the editorial team into proper bibliographic references in the [published Version of Record](#)
- Announced 12 November 2020 ([eLife post](#), [SWH announcement](#))

Software Heritage is a game changer

- **universal** archive
 - **all** the source code
 - yes, even 20 lines snippets/scripts for data mangling
 - more importantly: all the code your code depends upon!
- SWHID: uniform **intrinsic** identifier
 - independent of the development platform or technology
 - full guarantee of integrity (climate data anecdote goes here)

- 1 Introduction
- 2 (Bad) SOTA
- 3 Assessing the needs and a strategy to address them
- 4 Meet Software Heritage: universal archive of source code
- 5 Demo time!
- 6 Practical tools for researchers**
- 7 Shedding some light on the reference/identifiers confusion
- 8 Actions

Archive in one click: the *UpdateSWH* browser extension

Check and refresh the archive from your repository page

- for Firefox, Chrome and Edge; works on GitHub, GitLab, Bitbucket, Gitea, Codeberg
- a colour-coded button shows the archival status of the page you are viewing

What the button does

- **grey** — not archived yet → click to archive it
- **yellow** — outdated → click to trigger an update (Save Code Now)
- **green** — up to date → click to open the archive / copy the SWHID

no account, no setup: install and click

Archive automatically: forge *webhooks*

Preserve every important version with *zero* ongoing effort

Add *one* webhook URL in your repository settings:

`https://archive.softwareheritage.org/api/1/origin/save/webhook/gith`

(also `.../gitlab/` and `.../bitbucket/`)

- recommended: trigger on *tag, branch or release creation* — not on every push
- supported: GitHub, GitLab (and instances), Bitbucket, Gitea, SourceForge

see the [Software Heritage webhooks guide](#)

Describe your software: *CodeMeta* and CITATION.cff

Machine-readable metadata that travels *with* the code

- `codemeta.json` — a JSON-LD schema for software (authors, license, references, funding, ...), Software Heritage uses it for the multiple granularity citation button
richer than CITATION.cff
- `CITATION.cff` — the lightweight cousin; GitHub renders it as a “*Cite this repository*” button, Software Heritage uses it for the multiple granularity citation button
can be generated from CodeMeta

No JSON by hand: the CodeMeta generator

- fill in a web form → download a ready `codemeta.json`
codemeta.github.io/codemeta-generator
- drop both files at the root of your repository → archived *and* indexed by Software Heritage

A dedicated citation style

(in TeXLive; used by ACM templates since 2022)

Four entry types – `@software`, `@softwareversion`, `@softwaremodule`, `@codefragment` – carrying a `swhid` field, chained with `crossref`.

A worked example

```
@softwareversion{parmap-1.1.1,  
  author   = {Di Cosmo, R. and Danelutto, M.},  
  title    = {The Parmap library}, version = {1.1.1}, license = {LGPL-2.0},  
  swhid    =  
  {swh:1:rel:373e26...db773;origin=https://github.com/rdicosmo/parmap}}
```

the `swhid` makes the citation *reproducible*, long after URLs break

- 1 Introduction
- 2 (Bad) SOTA
- 3 Assessing the needs and a strategy to address them
- 4 Meet Software Heritage: universal archive of source code
- 5 Demo time!
- 6 Practical tools for researchers
- 7 Shedding some light on the reference/identifiers confusion**
- 8 Actions

Extrinsic vs Intrinsic identifiers

In a nutshell

(for more info see [this dedicated blog post](#))

Main difference: how the *relation* between *identifier* and *designated object* is created and maintained. *Persistence* is a key desired property.

	Extrinsic	Intrinsic
relation	register	convention
persistence	external ¹	internal
pre-internet	passport number, ISBN, SSN, etc.	Music/Chemistry notations <i>e.g. NaCl is table salt</i>
internet era	DOI, Handle, Ark, etc.	cryptographic hashes <i>e.g.: git, bitcoin, SWHID</i>

distributed software development (e.g. git) relies on *intrinsic identifiers*
uniform intrinsic identifier

we need a

¹"persistence... is a function of *administrative care*" [RFC 3650 \(Handle System Overview, 2003\)](#)

A DOI pointing to a Zenodo *zip*

- resolves to a **snapshot** you must *trust* — no built-in integrity check
- one opaque blob: no *history*, no *diff*, nothing addressable *inside* it
- granularity stops at "the whole deposit"

A *SWHID*

- **intrinsic**: the identifier *is* the cryptographic hash → self-verifying
- addresses *any* level: snapshot / revision / directory / file / **lines**
- works for *any* VCS (git, hg, svn) and for plain tarballs

a DOI answers "*which deposit?*"; a SWHID answers "*exactly which bytes, and prove it*"

- 1 Introduction
- 2 (Bad) SOTA
- 3 Assessing the needs and a strategy to address them
- 4 Meet Software Heritage: universal archive of source code
- 5 Demo time!
- 6 Practical tools for researchers
- 7 Shedding some light on the reference/identifiers confusion
- 8 Actions**

What this means for a (PhD student/young) researcher

Starting with your next paper

- **archive** the exact code behind every result (Save Code Now / deposit / webhook)
- **reference** claims at the right granularity: a SWHID for the revision, **directory**, the file, the lines
- **describe** it: a `codemeta.json` (and a generated `CITATION.cff`) in every repository
- **cite and credit** it in the bibliography, like any other research output, use and promote biblatex-software

As a reviewer or AEC member

- ask authors for *archival proof* (a SWHID), not a fragile URL
- one habit, applied every time, changes the norms of your community

The double-blind problem

Double-blind review needs your code — *without* revealing who wrote it.

`git-anonymize` : git-only, error-prone, leaks commit logs, makes a copy

A simpler, intrinsic answer

Share a *directory* SWHID of your code tree: `swh:1:dir:fdf3...a91c`

- **no identity**: hash *content alone* — no author, no origin, nothing to scrub
- **any technology**: a plain directory or tarball, not just git (Hg, SVN, no-VCS: all work)
- **zero overhead**: one command; reviewers can verify the bytes are exactly yours

```
swh identify --type directory ./ →  
swh:1:dir:fdf3...a91c
```

don't treat your code like an afterthought: *record, share and value it*
the tools are ready — starting with your next paper

Explore deeper in the full doctoral course notes at



References

-  UNESCO, *Draft recommendations on Open Science 2021*, ([online](#))
-  French Ministry of Research, *Second National Plan for Open Science 2021*, ([online](#))
-  EOSC SIRS Task Force, *Scholarly Infrastructures for Research Software* 2020, Publications office of the European Commission, ([10.2777/28598](#))
-  R. Di Cosmo, *Archiving and Referencing Source Code with Software Heritage* International Conference on Mathematical Software 2020 ([10.1007/978-3-030-52200-1_36](#))
-  J.F. Abramatic, R. Di Cosmo, S. Zacchiroli, *Building the Universal Archive of Source Code* CACM, October 2018 ([10.1145/3183558](#))

Appendix

- 9 FAIR
- 10 HAL and SWH
- 11 Broader policy issues
- 12 Detailed addressing ARDC
- 13 Revolutionary infrastructure, scientific challenges
- 14 All the source code
- 15 Milestones, policy
- 16 Public code, mirrors

What about FAIR? (Findable, Accessible, Interoperable, Reusable)

FAIR data principles *for data*

in a nutshell: metadata, metadata, metadata all over the place (makes sense for data)

But software is *not data* ...

the terms *interoperability* and *reusability* have precise technical meaning for software, and *differ significantly* from what is intended by the I and R of FAIR;

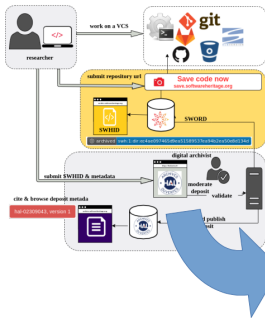
- see the entries for [software interoperability](#) and [software reusability](#)
- it is *very difficult* to achieve these properties even for commercial software developed by multi billion dollars corporations

FAIR for software is a distraction

let's focus on the real issues at stake: ARDC a good starting point

- 9 FAIR
- 10 HAL and SWH
- 11 Broader policy issues
- 12 Detailed addressing ARDC
- 13 Revolutionary infrastructure, scientific challenges
- 14 All the source code
- 15 Milestones, policy
- 16 Public code, mirrors

Overview of the Software Heritage / HAL synergy



<https://hal.archives-ouvertes.fr/hal-02130801>



Free and accessible knowledge

[Home](#) |
 [Submit](#) |
 [Browse+](#) |
 [Search](#) |
 [Documentation](#)

[hal-01030851 - version 1](#)

LinBox

[The LinBox Group](#)

- 1 ECD - Exact Computing
(SRM) - Laboratoire d'Informatique de la Robotique et du Mécatronique du Montpellier
- 2 ARIC - Arithmetic and Computing
- 3 Inria Grenoble - Rhône-Alpes, LIP - Laboratoire de l'Informatique du Parallélisme
- 4 ANALON - Algorithms and Software Architectures for Distributed and HPC Platforms
- 5 Inria Grenoble - Rhône-Alpes, LIP - Laboratoire de l'Informatique du Parallélisme
- 6 CIS - Department of Computer and Information Sciences [Newark]
- 7 COS - Department of Computer and Information Sciences [Newark]
- 8 INCU - Department of Mathematics [Raleigh]
- 9 United States Navy Academy
- 10 SCS - Symbolic Computation Group
- 11 CASC - Calcul Algébrique et Géométrie, Sécurité, Systèmes Complexes, Codes et Cryptologie
- LIR - Laboratoire Jean Kuntzmann

Abstract. LinBox is a C++ template library of routines for solution of linear algebra problems including linear system solution, rank determination, minimal polynomial, characteristic polynomial, and Smith normal form. Algorithms are provided for matrices with integer entries or entries in a finite field. A number of matrix storage types is provided, especially for blockdense representation of sparse or structured matrix classes. A few algorithms for rational matrices are available. Linear also uses underlying data structures and algorithms for integer, rational, polynomial, finite fields and rings, as well as dense and sparse matrix formats coming from the Givaro (<http://gforge.inria.fr/projects/givaro/>) and FFLAS-FFPACK (<http://bitbucket.org/fflas/ffpack>) libraries.

METADATA

version:	1.6.1
-----------------	--------------

Software License

GNU Lesser General Public License v2.1

Programming Language

C++

Code Repository

<https://github.com/LinBox-team/linbox>

COLLECTIONS

END-YTON | CNRS | IRRMM | ECD | UNIV-LYON1 | IMERS | INRIA | LINPAC-INTEL | LIR | LIR_MAD | LIR-MAD-CASC | UCL | UGA | AVR

CITATION

The LinBox Group. LinBox. 2010. (swat-1-der300). doi:[10.24303/swat-1-der300](https://doi.org/10.24303/swat-1-der300).
<https://arxiv.org/archive/other/arXiv/1210.0801>; view:
<http://www.lirmm.fr/~linbox/linbox-1.6.1.pdf>
 48:val=4811; linbox=swat-1; view=1; swat=1; swat=2000;
 7575:swat=2000; 119636:1486107; pdf=0. (hal-01030851)

Document type Article

Domin **Computer Science [cs]**

Computer Science [cs] > Symbolic Computation [cs.BC]

Complete list of metadata ▶ Show

BROWSE

Software Engineering

swat-1-der300:611a1424832bf65968f76250237fa1a595; origin=http://hal.archives-ouvertes.fr/hal-01030851; view=1; smp=19c296a80b6; 2020:75w/linbox=swat-1; val=4811; linbox=swat-1; view=1; swat=1; swat=2000; 660:75w/linbox=swat-119636:1486107; pdf=1

◀ Choose

EXPORT

Download	Bibtex	Tel	DC	XQuery

Enter a SWHID to resolve or keyword(s) to search for it

<https://hal.archives-ouvertes.fr/hal-02130801>

14 June 2019, 13:43 UTC

< Code Branches () Releases (0) Vists

Revision: e8e18328952266b7875c692963b11963b1496107 393bd11 / linbox-1.6.3 / linbox / config-blas.h Raw File

Tip revision: e8e18328952266b7875c692963b11963b1496107 authored by Software Heritage on 11 June 2019, 08:12 UTC
hal: Deposit 297 in collection hal

config-blas.h

```

1 /* config-blas.h
2 * Copyright (C) 2005 Pascal Giorgi
3 *      2007 Clement Pernet
4 * Written by Pascal Giorgi <pgiorgi@waterloo.ca>
5 *
6 * =====LICENCE=====
7 * This file is part of the library LinBox.
8 *
9 * LinBox is free software: you can redistribute it and/or modify
10 * it under the terms of the GNU Lesser General Public
11 * License as published by the Free Software Foundation; either
12 * version 2.1 of the License, or (at your option) any later version.
13 *
14 * This library is distributed in the hope that it will be useful,
15 * but WITHOUT ANY WARRANTY; without even the implied warranty of
16 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
17 * Lesser General Public License for more details.
18 *
19 * You should have received a copy of the GNU Lesser General Public
20 * License along with this library; if not, write to the Free Software
21 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
22 * =====LICENCE=====

```

#endif LINBOX config blas h

[swh:1:dir:393b611a1424f032e83569bf6762502371cfcf65](#)

Growing adoption of SWH in Academia (selection)

HAL software curated deposit workflow

Curated Archiving of Research Software Artifacts

International Journal of Digital Curation, 2020

Reference archive for swmath.org



See *code* links, e.g.

SemiPar package

IPOL (image processing)



- archive (deposit)
- reference
- BibLaTeX

eLife (life sciences)



- archive (save code now)
- reference

JTCAM (mechanics)

- [instructions for authors](#)
- biblatex-software in journal L^AT_EX class

Policy: France



*National Plan
for Open Science
and Research
Infrastructures*

Policy: Europe



EOSC SIRS report

- SWHIDs
- archive

Guidelines



Software Heritage

- 1 Prepare your public repository
Readme, authors & license files
- 2 Save your code
<http://save.softwareheritage.org/>
- 3 Reference your work
(full repository, specific version or code fragments)

- [summary](#)
- [ICMS 2020](#)

Call to action on ARDC: let's foster adoption!

Train students and colleagues to [archive and reference relevant source code](#)

- full details in the [ICMS 2020](#) article
- short operational [HOWTO online](#)

Engage conferences, journals, learned societies to use Software Heritage and SWHIDs

APIs for [save code now](#) and [deposit](#) are available to integrate with

- Research Articles
- Artifact Evaluation Committees
- Badging initiatives

Help grow and structure the community

- Promote the [ambassador program](#)
- Encourage our institutions to
 - include Software Heritage in their Open Science policy
 - become [member/sponsor](#)
 - build a Software Heritage mirror (see ENEA)

- 9 FAIR
- 10 HAL and SWH
- 11 **Broader policy issues**
- 12 Detailed addressing ARDC
- 13 Revolutionary infrastructure, scientific challenges
- 14 All the source code
- 15 Milestones, policy
- 16 Public code, mirrors



Ref.: CL/4363

Subject: Draft text of the UNESCO Recommendation on Open Science

Madam/Sir,

At its 40th session in November 2019, the UNESCO General Conference decided to elaborate a draft Recommendation on Open Science.

This was a major decision, which has since mobilized the entire Organization and all of its Member States in the development of this new standard-setting instrument.

After two years of joint work, this process is now entering its final phase, following the consensus reached on the draft text during the intergovernmental meeting of experts held from 6 to 11 May 2021.

I have the pleasure to submit to you this draft recommendation, which will be put forward for adoption at our next General Conference in November 2021.

The definitions and principles that it contains constitute a common – and currently unprecedented – framework to support scientific cooperation and make science more transparent, more accessible, more equitable and more inclusive.

For any further information, Sharmila Nair-Bedouelle, Assistant Director-General for Natural Sciences, is at your disposal at the following email address: openscience@unesco.org.

Thanking you for your commitment, please accept, Madam/Sir, the assurances of my highest consideration.


Audrey Azoulay
Director-General

Enclosure: 1

1. Draft text of the UNESCO Recommendation on Open Science

cc: Permanent Delegations to UNESCO
National Commissions for UNESCO

7, place de Fontenay
92626 Paris 15, France
Tel.: +33 (0)1 55 36 36 32
www.unesco.org

To Ministers responsible for relations with UNESCO

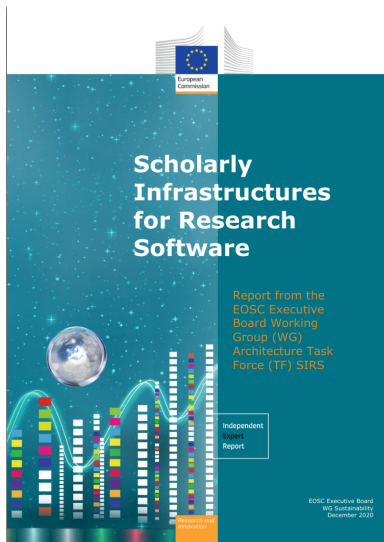
Selection from the recommendations

- Open Source for Open Science

"The source code must be included in the software release and made available on openly accessible repositories and the chosen license must allow modifications, derivative works and sharing under equal or compatible open terms and conditions"

- Infrastructures

"Open science infrastructures should be organized and financed upon an essentially not-for-profit and long-term vision, which enhance open science practices and guarantee permanent and unrestricted access to all, to the largest extent possible."

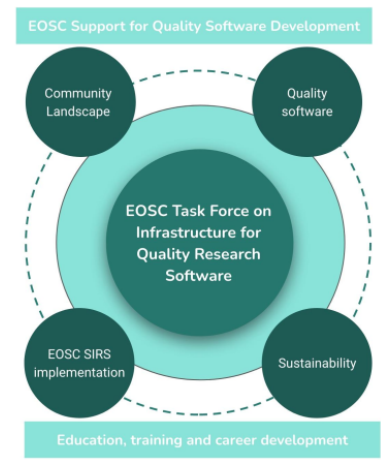


Important *policy tool* in Open Science (Dec 2020)

- 9 infrastructures
 - 3 archives
 - 3 open access publishers
 - 3 aggregators
- recommendations
 - archive in Software Heritage, use SWHID
 - open non profit
 - default to open source for research software

"all research software should be made available under an Open Source license by default, and all deviations from this default practice should be properly motivated"

See <https://doi.org/10.2777/28598>



Ongoing action in the EOSC

Task force on infrastructures for quality research software

- Foster the development and deployment of tools and services that allow researchers to properly archive, reference, describe with proper metadata, share and reuse research software.
- Improve the quality of research software, both from the technical and organizational point of view ...
- Increase recognition to software developers and maintainers of research software ...

See [the charter of the task force](#).

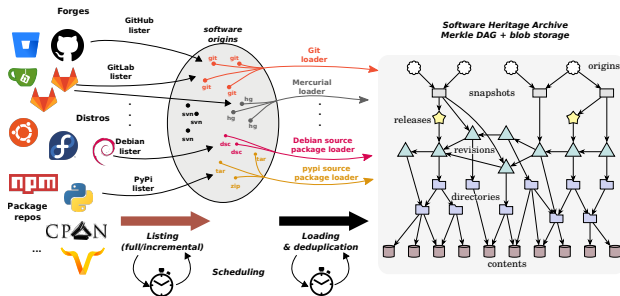
- 9 FAIR
- 10 HAL and SWH
- 11 Broader policy issues
- 12 Detailed addressing ARDC**
- 13 Revolutionary infrastructure, scientific challenges
- 14 All the source code
- 15 Milestones, policy
- 16 Public code, mirrors

Addressing the A(rchive)

Universal source code archive

not only research

(16B+ files, 260M+ projects)



- your research software *is likely there already!*
- anyone can trigger archival with save.softwareheritage.org
- selected partners can push to the archive via deposit.softwareheritage.org

Saving 250.000 endangered repositories...

- summer 2019: BitBucket announce Mercurial VCS phase out
- fall 2019: Software Heritage teams up with Octopus (funded by NLNet, thanks!)
- july 2020: BitBucket erases 250.000 repositories
- august 2020: bitbucket-archive.softwareheritage.org is live

... preserving the web of knowledge

(original tweet is [here](#))



Gabriel Altay
@gabrielaltay

Just realized @Bitbucket disabled all mercurial repositories when the @asclinet informed me that a link associated with an old paper of mine was down. Thought all was lost, but someone archived all the repos! very classy move by @octopus_net and @SWHeritage.

[Traduire le Tweet](#)

1:48 AM · 31 août 2020 · Twitter Web App

Bottomline

explicit deposit is important, ...

... and we must promote it...

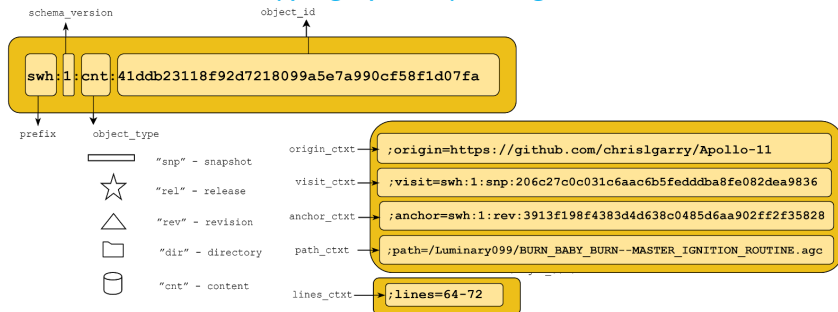
... but will never be enough.

(think also of all software dependencies!)

Software Hash Identifiers (SWHID)

see swhid.org

50+B intrinsic, decentralised, cryptographically strong identifiers, SWHIDs

In [SPDX 2.2](#); IANA "swh: "; WikiData [P6138](#); standard [ISO 18670](#)Full fledged *source code references* for traceability, integrity and reproducibilityExamples: [Apollo 11 AGC](#), [Quake III rsqrt](#); Guidelines available: [HOWTO](#) and [ICMS 2020](#)

Addressing D(escribe) and C(ite) in ARDC (see [ICMS 2020](#) for details)

Describe

- Collect *intrinsic metadata*
- Contributed the [Codemeta generator](#)

CodeMeta generator

Most fields are optional. Mandatory fields will be highlighted when generating Codemeta.

The software itself

Name

the software title

Description

Creation date

First release date

Cite/Credit

- Contributed *software citation style*
[biblatex-software](#), v 1.2-2 now on [CTAN](#)



- 9 FAIR
- 10 HAL and SWH
- 11 Broader policy issues
- 12 Detailed addressing ARDC
- 13 Revolutionary infrastructure, scientific challenges**
- 14 All the source code
- 15 Milestones, policy
- 16 Public code, mirrors

Summing up: a revolutionary infrastructure *designed for source code*



Software Heritage

www.softwareheritage.org

global source code archive

Library of Alexandria of source code



- harvest *all* software source code
- *on demand* harvesting and *curated deposit*

universal intrinsic identifiers

SWHID standard is independent of version control systems

uniform data model, *full graph* of development history

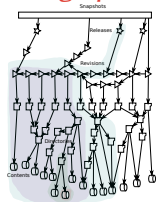
enables large scale, big code research

infrastructure for Open Science

base layer for software source code in the *Open Science architecture*

A revolutionary infrastructure *designed for software source code*

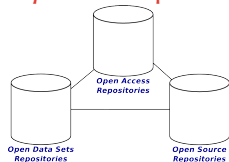
The *graph* of public software development



All software development
in a **single graph** ...

- enable traceability

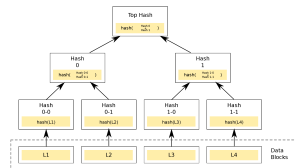
A *pillar* of Open Science



Reference **archive** of
Research Software

- reproducibility
- reference

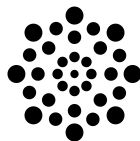
The *global ledger* of public code



... a **Merkle** graph

- ensure integrity

Reference platform for *Big Code* **uniform** data structure



- large scale studies
- cybersecurity
- machine learning, AI, ...

A challenging scientific and technical undertaking

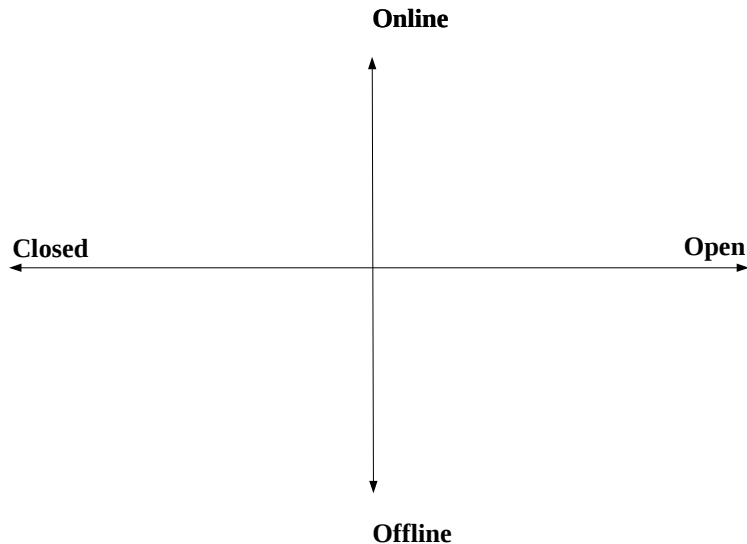
A novel, large infrastructure

- object storage with peculiar workload
- gigantic Merkle graph
- counting tens of billions of objects (reuse P. Flajolet's seminal work)
- and much more: see the 2021 technical roadmap

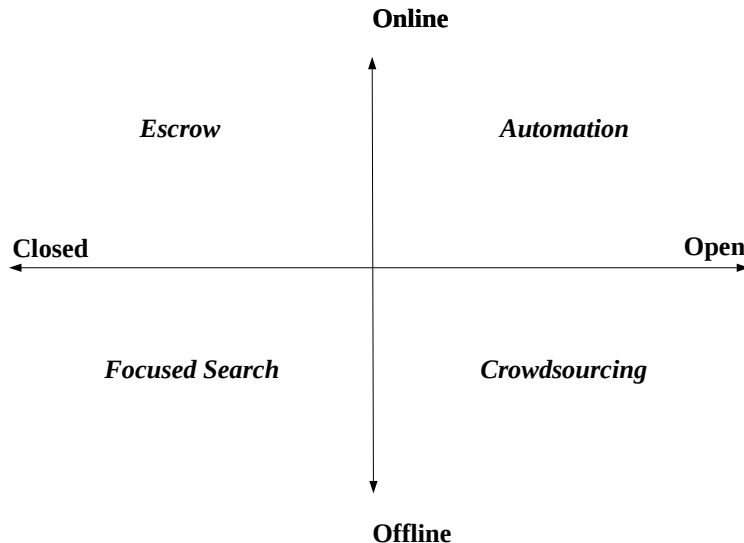
First datasets are available for Big Code analysis

- full graph of software development (~20Bn nodes, ~200Bn edges) see Pietri, Spinellis, Zacchiroli, MSR 2019
<https://dx.doi.org/10.1109/MSR.2019.00030>
- MSR 2020 mining competition see <https://2020.msrconf.org/track/msr-2020-mining-challenge#Call-for-Papers>

- 9 FAIR
- 10 HAL and SWH
- 11 Broader policy issues
- 12 Detailed addressing ARDC
- 13 Revolutionary infrastructure, scientific challenges
- 14 All the source code**
- 15 Milestones, policy
- 16 Public code, mirrors



All the source: strategies



- 9 FAIR
- 10 HAL and SWH
- 11 Broader policy issues
- 12 Detailed addressing ARDC
- 13 Revolutionary infrastructure, scientific challenges
- 14 All the source code
- 15 Milestones, policy**
- 16 Public code, mirrors

Summer 2015



The collection starts: first server, (very) early prototype

June 30th 2016



Public unveiling, with the first sponsors: Microsoft and DANS

April 3rd 2017



Unesco - Inria agreement on software access and preservation.

June 7th 2018



Opening the archive to the world

February 26th 2019



Publication of the expert meeting
**Paris Call on
Software Source Code**

June 2019



Software Heritage Graph as **Open Dataset on AWS**

"Upload filters": a threat to *all modern software development*

- developing platforms (GitHub, GitLab, Bitbucket, etc.)
- distribution platforms (Maven, Pypi, CRAN, CTAN, etc.)
- archives (Software Heritage)

We got an exclusion for

~~non for profit~~ open source software developing *and sharing* platforms

Key role of Software Heritage

policy-maker awareness, essential insights for NGOs, government contacts

- 9 FAIR
- 10 HAL and SWH
- 11 Broader policy issues
- 12 Detailed addressing ARDC
- 13 Revolutionary infrastructure, scientific challenges
- 14 All the source code
- 15 Milestones, policy
- 16 Public code, mirrors



code.etalab.gouv.fr (alpha)

Contact

Glossary

About

Etalab

Public sector source codes

This website lets you browse some of the source codes opened by public bodies. If your source code is not referenced on this website, [send us](#) a link to your repository.

Source code repositories

Organizations or groups

Figures

Free search

License

Language

☐ Forks only

☐ Hide archives

☐ With a description

☐ With known license

3669 repositories

Repository / group	Archive	Description	Updated	Forks	Stars	Issues
medle / SocialGouv		MedLé : plateforme permettant aux établissements de santé de déclarer leur activité médico-légale	11/8/2019	0	0	
reseauchaleur / deal-datalab			11/8/2019	0	0	

<https://code.etalab.gouv.fr>

Thomas Jefferson, February 18, 1791

...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.

Welcoming ENEA

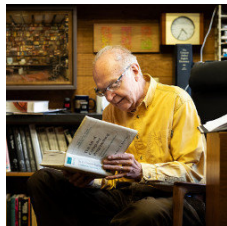


Italian National Agency for New Technologies,
Energy and Sustainable Economic Development

- first **institutional** mirror
- increased resilience
- **AI infrastructure** for researchers
- stepping stone to
an European joint effort

Calling for preservation: Donald Knuth and Len Shustek

Communications of the ACM, February 2021



"Telling historical stories is the best way to teach. It's much easier to understand something if you know the threads it is connected to."

Let's Not Dumb Down the History of Computer Science

Donald E. Knuth, Len Shustek

<https://doi.org/10.1145/3442377>

A unique opportunity

most of the creators are still here: we can talk to them!

but the clock is ticking...

Source code history for Security and Transparency

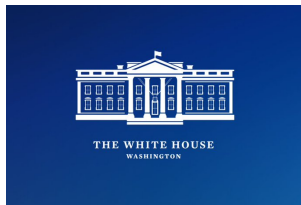
Where does reused software come from?



Do *you* know where it comes from?

- the software you ship
- the software you use
- the software you acquire
- the software that
 - has that bug
 - has that vulnerability

KYSW: Know Your SoftWare



Like KYC in banking, KYSW is now essential all over IT...

Sec. 4. Enhancing Software Supply Chain Security

ensuring and attesting, to the extent practicable, to the integrity and provenance of open source software

May 2021 POTUS Executive Order