Software Heritage

a revolutionary infrastructure for Open Source and Open Science

Roberto Di Cosmo
Director, Software Heritage
Inria and Université Paris Cité

March 2025

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

1. Introduction
2. (Open) Source for Open Science
3. Software Heritage in the big picture
4. Demo time
5. Much more than an archive
6. Conclusion

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

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)

```
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
```

Covid Sim (excerpt)

```
/**
 * @brief The basic unit of the simulation and is associated to a geographical location.
 *
 * Interventions (e.g., school closures) are tracked at this level. It contains a list of its
 * members (people), places (schools, universities, workplaces etc.), road networks, links to
 * airports etc.
 */
struct Microcell
{
        /* Note use of short int here limits max run time to USHRT_MAX*ModelTimeStep - e.g. 65536*0.25=16384 days=44 yrs.
           Global search and replace of 'unsigned short int' with 'int' would remove this limit, but use more memory.
        */

        int n; // Number of people in microcell
        int adunit; // admin unit microcell belongs to
        int* members; // array of members/hosts of microcell

        int* places[MAX_NUM_PLACE_TYPES]; // list of places (of various place types) within microcell
        unsigned short int NumPlacesByType[MAX_NUM_PLACE_TYPES]; // number of places (of various place types) within mircocell
        unsigned short int keyworkerproph, move_trig, place_trig, socdist_trig, keyworkerproph_trig;
        unsigned short int move_start_time, move_end_time;
        unsigned short int place_end_time, socdist_end_time, keyworkerproph_end_time;
        TreatStat moverest, treat, vacc, socdist, placeclose;
        unsigned short int treat_trig, vacc_trig;
        unsigned short int treat_start_time, treat_end_time;
        unsigned short int vacc_start_time;
        IndexList* Airportlist;
};
```

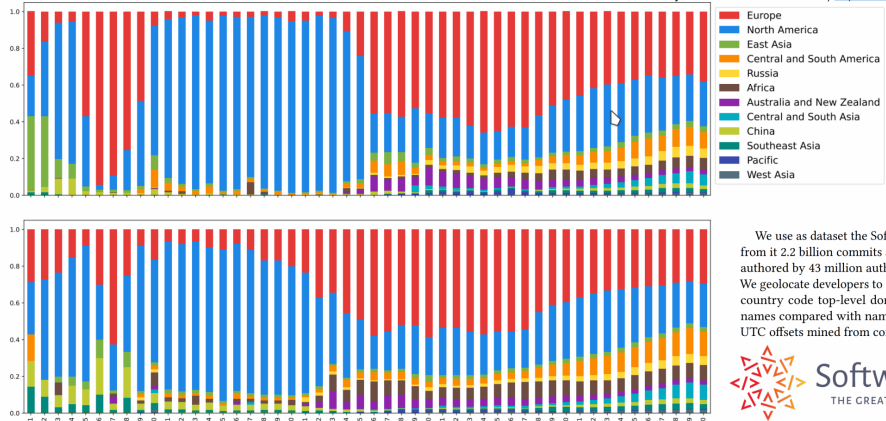
Len Shustek, Computer History Museum 2006

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

(Open) Source Code comes from all over the world

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA Davide Rossi and Stefano Zacchiroli

Geographic Diversity in Public Code Contributions: An Exploratory Large-Scale Study Over 50 Years. MSR 2022) https://doi.org/10.1145/3524842.3528471

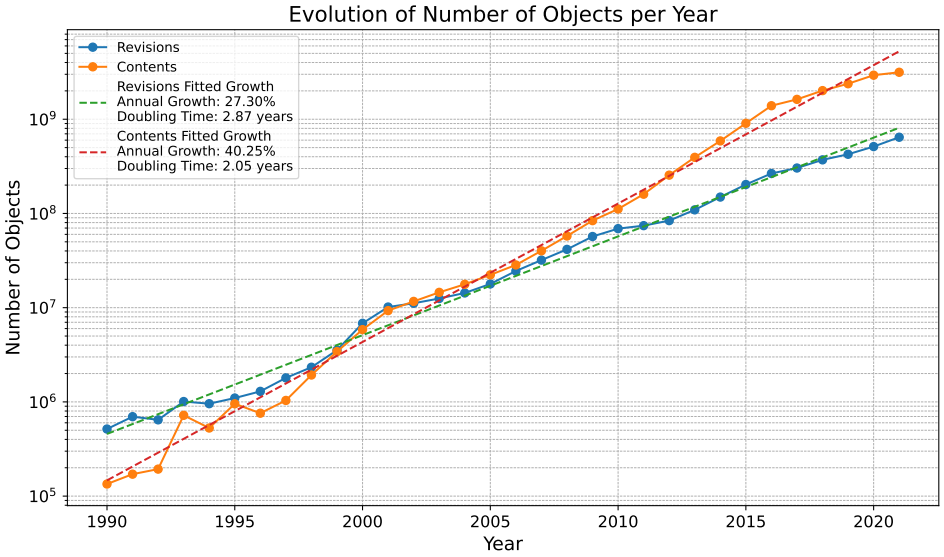


We use as dataset the Software Heritage archive [3] and analyze from it 2.2 billion commits archived from 160 million projects and authored by 43 million authors during the 1971–2021 time period. We geolocate developers to 12 world regions, using as signals email country code top-level domains (ccTLDs) and author (first/last) names compared with name distributions around the world, and UTC offsets mined from commit metadata.

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Figure 3: Ratio of commits (above) and active authors (below) by world zone over the 1971–2020 period.

(Open) Source Code grows at an exponential rate



Outline

1 Introduction

2 (Open) Source for Open Science

3 Software Heritage in the big picture

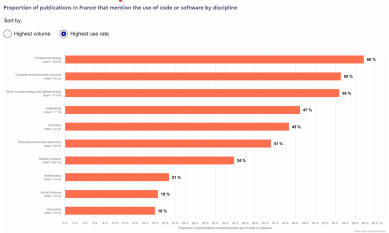
4 Demo time

5 Much more than an archive

6 Conclusion

Software Source Code: pillar of Open Science

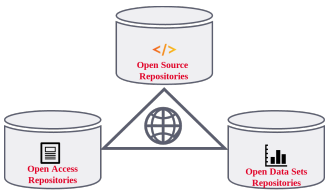
Software powers modern research



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!

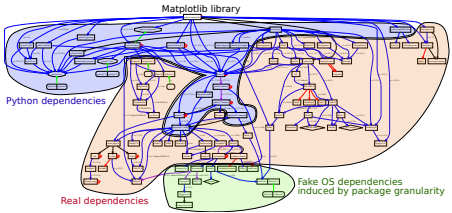
Source code is *special* (software is *not* data)

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



Precious, endangered *executable* and *human readable* knowledge

key people passing away, platforms (GoogleCode, Gitorious, etc.) closing down …

no organised effort to catalog and archive it

Policy on the rise

Paris Call on Software Source code (2019, UNESCO)

40 international experts call to *"promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, […] recognising in the careers of academics their contributions to high quality software development, in all their forms"*

UNESCO recommendations for Open Science, 2018-2021

"The source code must be included in the software release and […] the license must allow modifications, derivative works and sharing […]"
"Open science infrastructures should be […] essentially not-for-profit and long-term"

EOSC SIRS report: Software Source Code and Open Science, 2020

connect scholarly ecosystem via Software Heritage

And much more

Software track in OSEC 2022, Software working group launched in Science Europe, DFG adds software to model CV (9/22), NASA unveils Open Science policy (12/22), …

French National plan for Open Science, 2021-2024

MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR,
DE LA RECHERCHE
ET DE L'INNOVATION

Liberté
Égalité
Fraternité

SECOND FRENCH PLAN FOR OPEN SCIENCE

Generalising open science in France 2021-2024

1

Launch on 6 July 2021 by Frédérique Vidal, Minister for Higher Education, Research and Innovation

- Multiplying the **levers for change** in order to **generalise open science practices**
- Structuring the **policy for opening up or sharing research data**
- New commitments to the **opening of source code** produced by research
- **European and international inclusion** in the context of the French Presidency of the European Union
- **Disciplinary and thematic variations**: open science policies must be adapted to disciplinary specificities

2

Path Three :

Opening up and promoting source code produced by research

7 **Recognize and support the dissemination under an open source license of software** produced by publicly funded research programmes

8 **Highlight the production of source code** from higher education, research and innovation

9 Define and promote an **open source software policy**

« The opening of software source code is a major challenge for the **reproducibility** of scientific results. »

« Distribution of software products under **open source licence** will be preferred. »

3

Define and promote an open source software policy

- Produce a **National Charter for Open Source Software** coming from higher education, research and innovation
- Develop the **link between data and software** through a network of **Chief Data Officers** in the various universities and research performing organisations.
- Develop the **economic models of open source software** and make them known within commercialization services
- **Support Software Heritage** and recommend it for the archiving and referencing of source code

Recognise source code as a contribution to research

- Create an **open source research software prize**
- **Provide greater recognition** for software production in the career of researchers, research support staff

Build an ecosystem that connects code, data and publications

- Develop **proper coordination** between software forges, open publication archives, data repositories and the scientific publishing sector.

4

Five action lines (see details online)

- Identifying and highlighting research software production
- Technical and social tools and best practices
- Valorization and sustainability
- Liaison and animation at national, European, and international levels
- Recognition and careers

Source Code primer

key concepts

- for students
- for teachers
- for researchers

Report on software forges

in academia (FR):

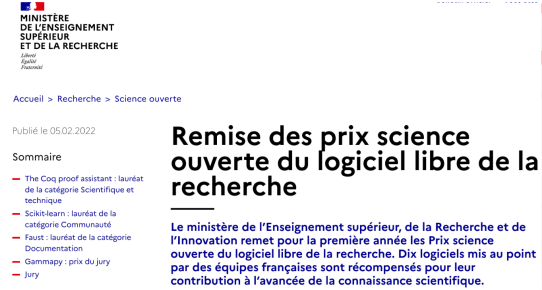
- needs
- options
- limitations

Annual award

Establishing a national research software award. Open Research Europe 2023

Software national award

First edition, 2022 prize


MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR
ET DE LA RECHERCHE

Accueil > Recherche > Science ouverte

Publié le 05.02.2022

Sommaire

- The Coq proof assistant : lauréat de la catégorie Scientifique et technique
- Scikit-learn : lauréat de la catégorie Communauté
- Faust : lauréat de la catégorie Documentation
- Gammapy : prix du jury
- Jury

Remise des prix science ouverte du logiciel libre de la recherche

Le ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation remet pour la première année les Prix science ouverte du logiciel libre de la recherche. Dix logiciels mis au point par des équipes françaises sont récompensés pour leur contribution à l'avancée de la connaissance scientifique.

- 129 projects
- 4 awards
- 6 accessit
- first edition

- Coq proof assistant
- Scikit-Learn ML/AI
- Faust music
- Gammapy astronomy

Second edition, 2023 prize


MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR
ET DE LA RECHERCHE

Accueil > Recherche > Science ouverte

Publié le 23.11.2023

Sommaire

- PlanQOOLIN : espoir de la catégorie « Scientifique et technique »
- Smilei : lauréat de la catégorie « Scientifique et technique »
- NoiseExplorer : espoir de la catégorie « Communauté »
- QCaml : lauréat de la catégorie « Communauté »
- KeOps : espoir de la catégorie « Documentation »
- Brian : lauréat de la catégorie « Documentation »
- Fink : espoir de la catégorie « Coup de cœur » du jury
- Hyphe : lauréat de la catégorie « Coup de cœur » du jury
- Jury

Remise des prix science ouverte du logiciel libre de la recherche 2023

Le ministère de l'Enseignement supérieur et de la Recherche remet pour la deuxième édition les Prix science ouverte du logiciel libre de la recherche. Huit logiciels développés par des équipes françaises sont récompensés pour leur contribution à l'avancée de la connaissance scientifique ou pour le caractère prometteur de leurs travaux.

- 66 projects
- 4 awards
- 4 "espoirs"
- now runs annually

What is at stake

ARDC

- Archive for retrieval (*reproducibility*)
- Reference for identification (*reproducibility*)
- Describe for discovery and reuse
- Cite/Credit for credit and evaluation

Before ARDC

- Development practices and tools (VCS, build system, test suites, CI, code quality, …)
- Opening up towards a community (documentation, organization, communication)

Need training, tooling, infrastructures, best practices

Beyond ARDC

- Policies (dissemination, reuse, careers, …)
- Sustainability (legal, financial, etc.)
- Technology transfer
- Advanced technologies and tools (quality, traceability, etc.)

let's focus on the ARDC part

How do we tackle preservation?

A - Since the ~~1970's~~ 1990's

.zip or .tar file on:

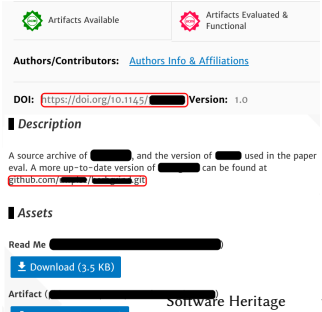
- ~~ftp server~~ (e.g. gnu)
- web page (example)
- document archive (+ DOI sample)

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!

Outline

1. Introduction
2. (Open) Source for Open Science
3. Software Heritage in the big picture
4. Demo time
5. Much more than an archive
6. Conclusion

Unveiled in 2016

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

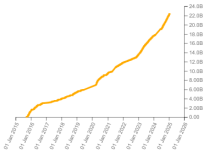
A *universal* software archive, as a shared infrastructure

One infrastructure
open and shared

The largest archive ever built

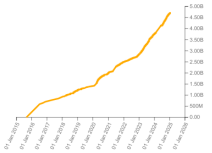
Source files

22,548,654,226



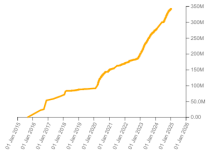
Commits

4,743,656,085



Projects

344,672,354



Directories

17,848,569,305

Authors

86,143,084

Releases

102,056,508

Bitbucket 2,578,475 origins	56,975 origins	git 33,350 origins
27,377 origins	debian 141,834 origins	88,561 origins
GitHub 249,868,189 origins	gitiles 24,378 origins	GitLab 5,736,223 origins
git 3,791 origins	Gogs 394 origins	1,887,337 origins
Guix 68,391 origins	GNU 354 origins	heptapod 1,340 origins
launchpad 654,755 origins	Maven 312,179 origins	NixOS 48,905 origins
npm 4,003,267 origins	5,438 origins	Packagist 376,882 origins

Sharing the vision

United Nations Educational, Scientific and Cultural Organization

Donors, members, sponsors

Diamond sponsors

Platinum sponsors

Gold sponsors

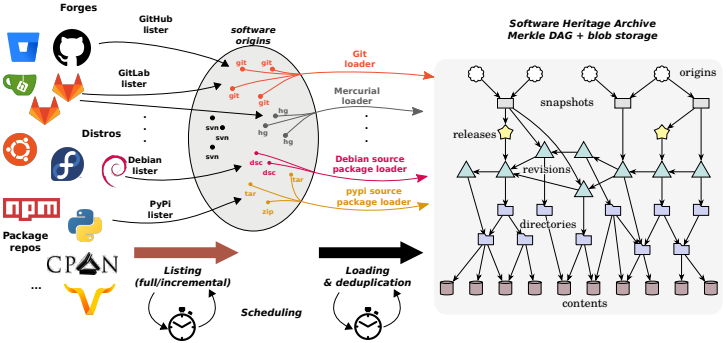
Silver sponsors

Bronze sponsors

And many more ...

www.softwareheritage.org/support/testimonials

The archive under the hood



Global development history permanently archived in a uniform data model

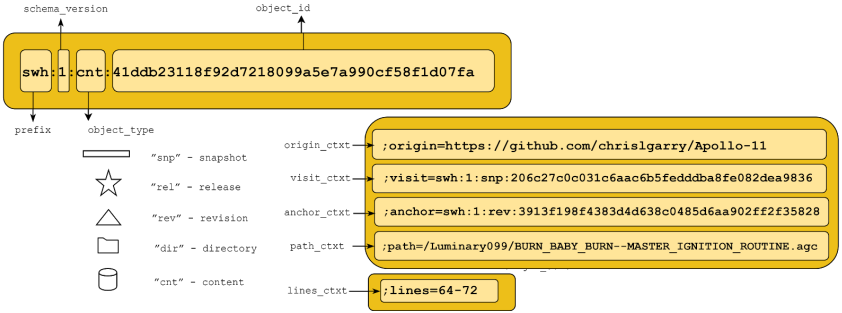
- over 22 billion unique source files from over 340 million software projects
- ~2PB (compressed) blobs, ~50 B nodes, ~800 B edges

The Software Hash persistent identifier (SWHID)

Software Hash Identifiers (SWHID)

see swhid.org

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



In SPDX 2.2; IANA "swh:"; WikiData P6138; ISO standardization ongoing DIS 18670

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

Examples: Apollo 11 AGC, Quake III rsqrt; Guidelines available: HOWTO and ICMS 2020

Outline

1. Introduction
2. (Open) Source for Open Science
3. Software Heritage in the big picture
4. Demo time
5. Much more than an archive
6. Conclusion

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

Addressing ARDC: an example is worth 1000 words

Software metadata: codemeta.json

- example from Parmap, created with Codemeta generator
- software citation (see detailed HOWTO) and biblatex-software

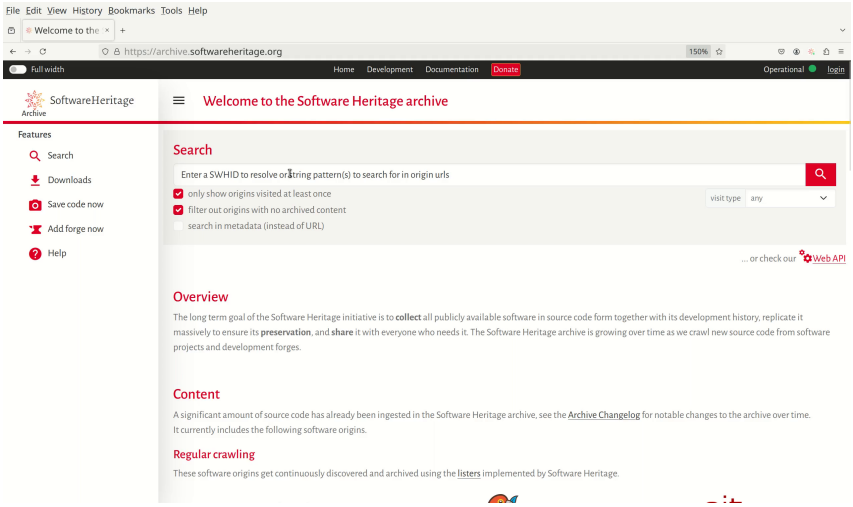
Integration with the HAL national french open access archive

- Curated deposit: metadata quality due to moderation
- examples: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …
- generation of reports, cv, web pages: for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools

Software Heritage + a *curated* metadata repository allows to address all needs …

- *researcher, engineer*: archival, reference, credit, CV etc. *with a little effort from them*
- *labs, organizations*: track and report software production, *curated* catalog

Get Swhid for a code snippet in the archive



video

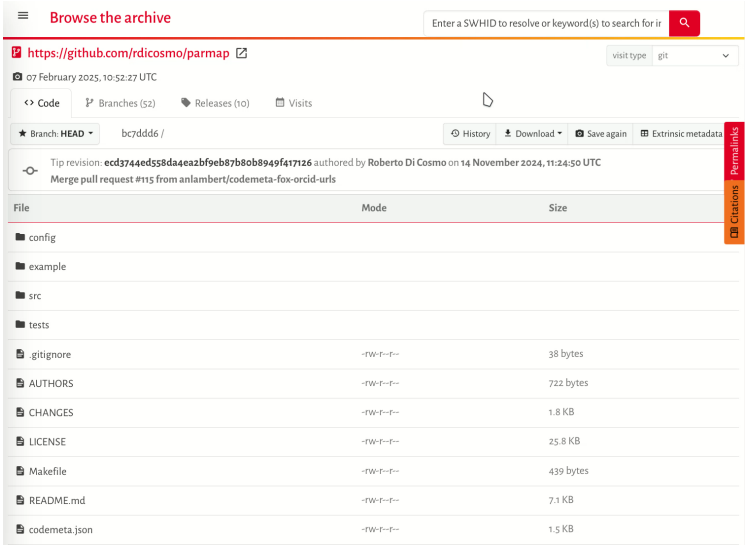
Spice up publications with Swhid


```
let simplemapper ncores compute opid al combine =
  (* init task parameters *)
  let ln = Array.length al in
  let chunksize = ln/ncores in
  (* create descriptors to mmap *)
  let fdarr=Array.init ncores (fun _ -> tempfd()) in
  (* spawn children *)
  for i = 0 to ncores-1 do
    match Unix.fork() with
      0 -> (* children code: compute on the chunk *)
           (let lo=i*chunksize in
            let hi=if i=ncores-1 then ln-1
                   else (i+1)*chunksize-1 in
            let v = compute al lo hi opid in
            marshal fdarr.(i) v;
            exit 0)
    | -1 -> failwith "Fork error"
    | pid -> ()
  done;
  (* wait for all children *)
  for i = 0 to ncores-1 do ignore(Unix.wait()) done;
  (* read in all data *)
  let res = ref [] in
  (* accumulate the results in the right order *)
  for i = 0 to ncores-1 do
      res:= ((unmarshal fdarr.((ncores-1)-i)):'d)::!res;
  done;
  (* combine all results *)
  combine !res;;
```

Figure 1: Simple implementation of the distribution, fork, and recollection phases in `Parmap` (slightly simplified from the actual code in the version of Parmap used for this article)

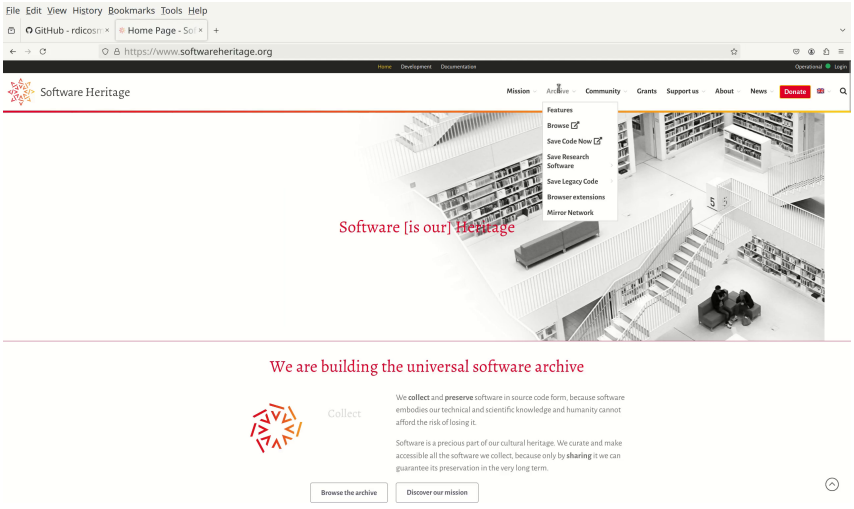
video

Get a citation for an archived code



video

Use the updateswh browser extension



video

Selected adoption indicators: users

From Melissa Harrison's OSEC 2022 talk

What are they "referencing"?

source	n	percentage
Not available	2868	46.22
GitHub	1151	18.55
software heritage	387	6.24
zenodo	142	2.29
r package	70	1.13
cran	56	0.90
r package version	54	0.87
gitlab	35	0.56

- 6205 "software" references identified
- Top 8 listed, then long tail of 1055 other sites – 932 are unique "source"

Use on replicabilitystamp.org

Lightweight Curvature Estimation on Point Clouds with Randomized Corrected Curvature Measures

Jacques-Olivier Lachaud, David Coeurjolly, Céline Labart, Pascal Romon, Boris Thibert

Wiley Computer Graphics Forum (CGF)

doi Repository

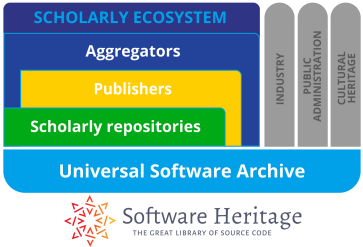
HAL+SWH in the Open Science software booklet

Funding agencies recommendations ANR 2023 guidelines (p. 17)

Enfin, conformément au 2ème Plan national pour la science ouverte, L'ANR recommande que les logiciels développés durant le projet soient mis à disposition sous une licence libre[30] et que les codes sources soient stockés dans l'archive Software Heritage[31] en indiquant la référence au financement ANR.

Selected adoption indicators: infrastructures

EOSC SIRS report: Software Source Code and Open Science, 2020



Connect scholarly ecosystem with the whole software ecosystem

See e.g. the French public administration open source catalog

Ongoing work: FAIRCORE4EOSC

A full workpackage:

- connectors with InvenioRDM (Zenodo), episcience, Dagstuhl, swMath, etc.
- Software Heritage mirror for the European Open Science Cloud (EOSC)
- standardisation of CodeMeta and SWHID

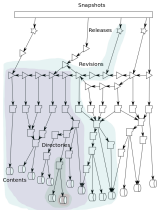
Outline

1. Introduction
2. (Open) Source for Open Science
3. Software Heritage in the big picture
4. Demo time
5. Much more than an archive
6. Conclusion

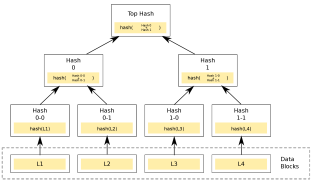
A revolutionary infrastructure

Modern "Library of Alexandria", *international, non profit, long term* initiative
addressing the needs of *industry, research, culture and society as a whole*

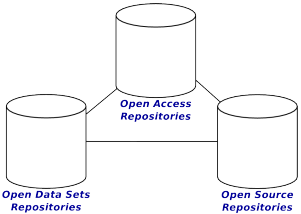
Software Graph



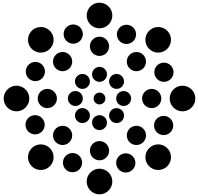
Software Blockchain



Open Science pillar



Big Code

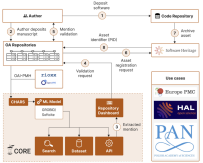


One infrastructure, *shared*: more efficient, less waste …

… for AI, Cybersecurity, Public Administration, Software Engineering … see UNESCO Symposium 2025

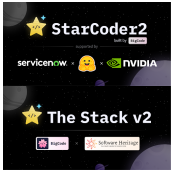
Software Heritage: 2024/2025 hilights

January 2024 — SoFAIR Kickoff



Massive identification, archival and reference of software in academic publications

February 2024 — First AI contact



StarCoder2 trained on GitHub subset of Software Heritage

February 2024 — SWHSec Kickoff

SWHSec: 5 years PTCC project
Software supply chain Security

Academic interoperability

EOSC
FAIR-IMPACT
FAIRCORE4EOSC

Connect with Zenodo, Epi– science, Dagsthul, SwMath, OpenAire, …

Fall 2024 — CodeCommons

Building the reference software knowledge base for AI

January 2025 UNESCO Symposium

Sponsors meeting, worldwide speakers

Outline

1. Introduction
2. (Open) Source for Open Science
3. Software Heritage in the big picture
4. Demo time
5. Much more than an archive
6. Conclusion

A unique opportunity

https://softwareheritage.org

Software Heritage is

- vendor neutral, open source
- worldwide, long term initiative

Software Heritage enables

- archival and reference for reproducibility
- cost mutualisation, global knowledge base

Call to action

- show it to students, engineers, researchers: they will adopt it
- include Software Heritage in your Open Science policy
- integrate it with your academic infrastructure
- join Software Heritage, support sustainability, steer development
- get involved with the open source infrastructure, develop new applications