

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

- 1 Introduction
- 2 Software Heritage in the big picture
- 3 Demo time
- 4 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

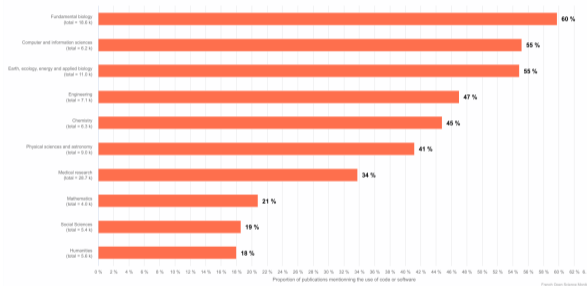
Software is a pillar of Open Science

Software powers modern research

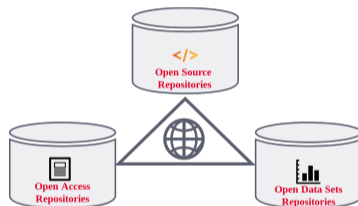
Proportion of publications in France that mention the use of code or software by discipline

Sort by:

Highest volume Highest use rate



Over 20% of articles using software across all disciplines share it
2024 French Open Science Monitor



Links are **important**

Nota bene

software may be a *tool*, a *research outcome* and a *research object*
access to the *source code* is essential!

Software Source Code is our 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     GOTOP00H     # 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 microcell
    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

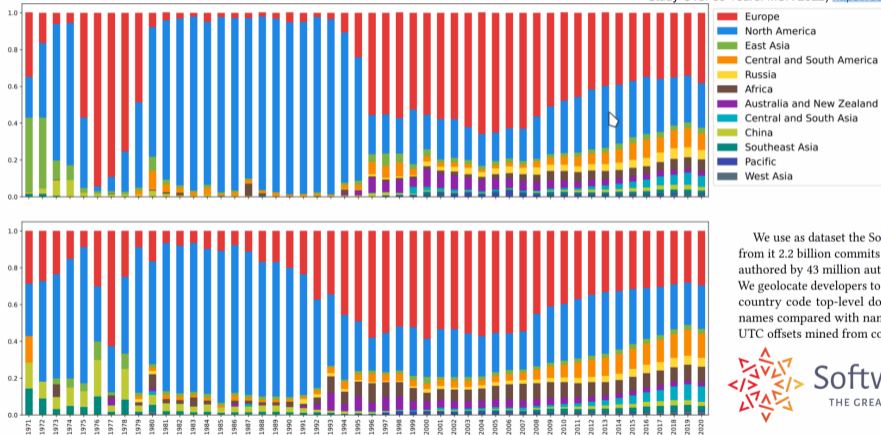
“Source code provides a view into the mind of the designer.”

(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>



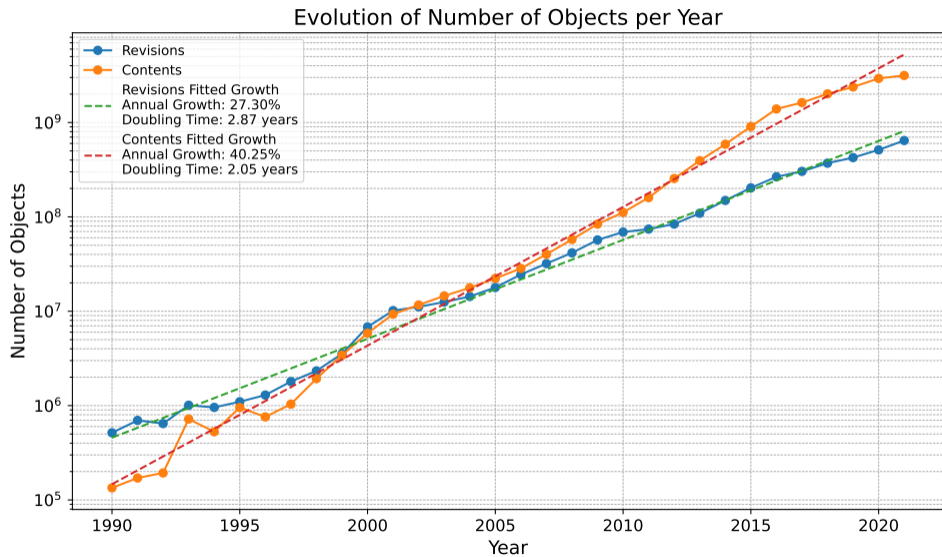
We use as dataset the Software Heritage archive [3] and analyze from it 2.2 billion commits archived from 160 million projects and authored by 43 million authors during the 1971–2021 time period. We geolocate developers to 12 world regions, using as signals email country code top-level domains (ccTLDs) 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



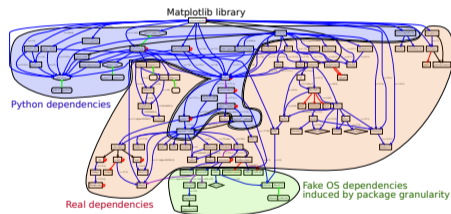
Source code is *special* (software is *not* data)

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

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.)

We need a *universal archive* to address ARDC: now we have one!

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

SECOND FRENCH PLAN FOR OPEN SCIENCE

Generalising open science in France 2021-2024



Second French Plan for Open Science



GENERALISING
OPEN SCIENCE
IN FRANCE 2021-2024

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

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

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

8

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

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

9

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

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.

- 1 Introduction
- 2 Software Heritage in the big picture**
- 3 Demo time
- 4 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



Software Heritage

www.softwareheritage.org

@swheritage

Contact: roberto@dicosmo.org

March 2025

12 / 26

Bitbucket 2,578,475 origins	debian 141,834 origins	git 33,350 origins
R 27,377 origins	Gogs 394 origins	GitLab 5,736,223 origins
GitHub 249,868,189 origins	Guix 68,391 origins	heptapod 1,340 origins
git 3,791 origins	GNU 354 origins	NixOS 48,905 origins
Maven 312,179 origins	npm 4,003,267 origins	Packagist 376,882 origins

Sharing the vision



And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors



Diamond sponsors



Platinum sponsors



Gold sponsors



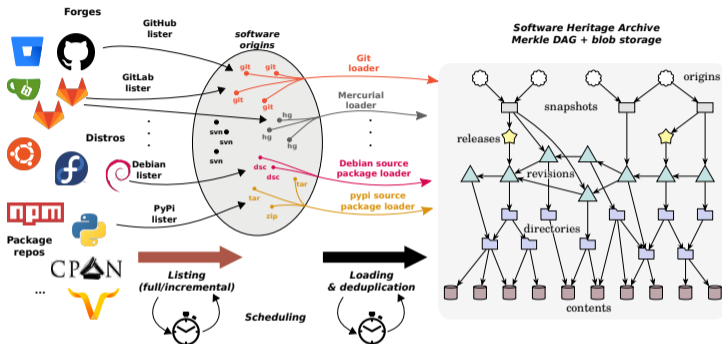
Silver sponsors



Bronze sponsors



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](#)

- 1 Introduction
- 2 Software Heritage in the big picture
- 3 Demo time**
- 4 Conclusion

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

The screenshot shows the Software Heritage website in a browser. The browser's address bar displays `https://archive.softwareheritage.org`. The website's navigation bar includes links for Home, Development, Documentation, and a red 'Donate' button. A 'Full width' toggle is visible on the left, and 'Operational' status and a 'login' link are on the right. The main content area features a search section with a text input field containing the placeholder 'Enter a SWHID to resolve or string pattern(s) to search for in origin urls'. Below the input are three checked checkboxes: 'only show origins visited at least once', 'filter out origins with no archived content', and 'search in metadata (instead of URL)'. A 'visit type' dropdown menu is set to 'any'. A red search button is on the right. Below the search section is an 'Overview' section with the text: 'The long term goal of the Software Heritage initiative is to collect all publicly available software in source code form together with its development history, replicate it massively to ensure its preservation, and share it with everyone who needs it. The Software Heritage archive is growing over time as we crawl new source code from software projects and development forges.' Below that is a 'Content' section with the text: 'A significant amount of source code has already been ingested in the Software Heritage archive, see the Archive Changelog for notable changes to the archive over time. It currently includes the following software origins.' The final section is 'Regular crawling' with the text: 'These software origins get continuously discovered and archived using the listers implemented by Software Heritage.'

video

Spice up publications with Swhid

```
5 / 11 | - 175% + | [ ] ↻
1 let simplemapper ncores compute opid al combine =
2   (* init task parameters *)
3   let ln = Array.length al in
4   let chunksize = ln/ncores in
5   (* create descriptors to mmap *)
6   let fdarr=Array.init ncores (fun _ -> tempfd()) in
7   (* spawn children *)
8   for i = 0 to ncores-1 do
9     match Unix.fork() with
10      0 -> (* children code: compute on the chunk *)
11          (let lo=i*chunksize in
12            let hi=if i=ncores-1 then ln-1
13                  else (i+1)*chunksize-1 in
14              let v = compute al lo hi opid in
15                marshal fdarr.(i) v;
16                exit 0)
17      | -1 -> failwith "Fork error"
18      | pid -> ()
19   done;
20   (* wait for all children *)
21   for i = 0 to ncores-1 do ignore(Unix.wait()) done;
22   (* read in all data *)
23   let res = ref [] in
24   (* accumulate the results in the right order *)
25   for i = 0 to ncores-1 do
26     res:= ((unmarshal fdarr.((ncores-1)-i)):'d)::!res;
27   done;
28   (* combine all results *)
29   combine !res;
```

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

video

Get a citation for an archived code

☰ Browse the archive 🔍

<https://github.com/rdicosmo/parmap> visit type: git

🕒 07 February 2025, 10:52:27 UTC

🔗 Code 🌿 Branches (52) 📦 Releases (10) 📅 Visits

★ Branch: HEAD - bc7ddd6 / 🕒 History 📄 Download 🔄 Save again 📄 Extrinsic metadata

🔑 Tip revision: **ecd3744ed558da4ea2bf9eb87b80b8949f417126** authored by Roberto Di Cosmo on 14 November 2024, 11:24:50 UTC
Merge pull request #115 from anlambert/codemeta-fox-orcid-urls

File	Mode	Size
config		
example		
src		
tests		
.gitignore	-rw-r--r--	38 bytes
AUTHORS	-rw-r--r--	722 bytes
CHANGES	-rw-r--r--	1.8 KB
LICENSE	-rw-r--r--	25.8 KB
Makefile	-rw-r--r--	439 bytes
README.md	-rw-r--r--	7.1 KB
codemeta.json	-rw-r--r--	1.5 KB

🔗 Citations 📄 Permalinks

video

Use the updateswh browser extension

File Edit View History Bookmarks Tools Help

GitHub - rdicosm... Home Page - Soft... +

https://www.softwareheritage.org

Software Heritage

Mission Archive Community Grants Support us About News Donate

- Features
- Browse
- Save Code Now
- Save Research Software
- Save Legacy Code
- Browser extensions
- Mirror Network

Software [is our] Heritage

We are building the universal software archive



Collect

We **collect** and **preserve** software in source code form, because software embodies our technical and scientific knowledge and humanity cannot afford the risk of losing it.

Software is a precious part of our cultural heritage. We curate and make accessible all the software we collect, because only by **sharing** it we can guarantee its preservation in the very long term.

[Browse the archive](#)

[Discover our mission](#)



video

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)

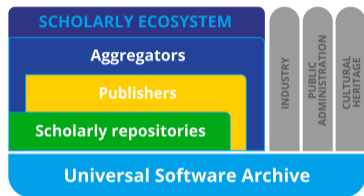


HAL+SWH in the Open Science software booklet

Funding agencies recommendations ANR 2023 guidelines (p. 17)

Enfin, conformément au 2^{ème} Plan national pour la science ouverte, L'ANR recommande que les logiciels développés durant le projet soient mis à disposition sous une licence libre³⁰ et que les codes sources soient stockés dans l'archive Software Heritage³¹ en indiquant la référence au financement ANR.

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

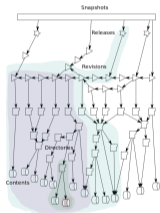
A full workpackage:

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

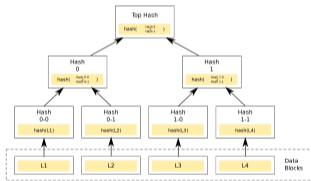
A revolutionary infrastructure

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

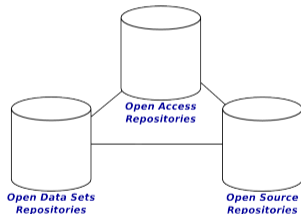
Software Graph



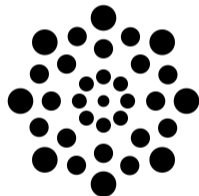
Software Blockchain



Open Science pillar



Big Code



One infrastructure, shared: more efficient, less waste ...

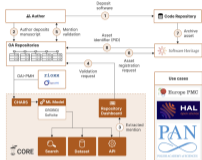
... for AI, Cybersecurity, Public Administration, Software Engineering ... see [UNESCO](#)

[Symposium 2025](#)

Software Heritage: 2024/2025 hilights

January 2024

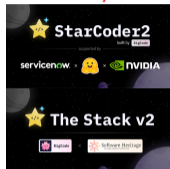
SoFAIR Kickoff



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



Connect with Zenodo, Epi-science, Dagstuhl, SwMath, OpenAire, ...

Fall 2024

CodeCommons



Building the reference software knowledge base for AI



January 2025 UNESCO Symposium



Sponsors meeting, worldwide speakers



Opportunity: your Institutional Portal!

Monitor and showcase your institution's software production



DALL-E's view of an institutional portal

- Curated metadata under institutional authority
- Persistent reference
- Uniform citation
- Automated extraction of reports and indicators

Why SWH?

- platform agnostic, metadata from multiple institutions
- benefit from Software Heritage's future developments

Get involved in the portal specification!

- curation workflow
 - researcher initiated (swhid deposit) vs institution initiated (metadata deposit)
- product deployment: on premise vs SaaS
- design of reports and extraction formats

- 1 Introduction
- 2 Software Heritage in the big picture
- 3 Demo time
- 4 Conclusion

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

Software Heritage



Annual Report 2024

Collecting, preserving and sharing software source code since 2015

