

# Software Heritage: key infrastructure for Open Science

Roberto Di Cosmo

Director, Software Heritage  
Inria and Université de Paris Cité

9 May 2023



Software Heritage  
THE GREAT LIBRARY OF SOURCE CODE

- 1 Introduction
- 2 Software and Open Science
- 3 An emerging policy framework
- 4 Assessing the needs and a strategy to address them
- 5 Meet Software Heritage (and the HAL connection)
- 6 Demo time!
- 7 Software Heritage, cont'd
- 8 Actions



# Short Bio: Roberto Di Cosmo

Computer Science professor in Paris, now working at INRIA

- 30+ years of research (Theor. CS, Programming, Software Engineering, Erdos #: 3)
- 20+ years of Free and Open Source Software
- 10+ 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](http://www.mancoosi.org)

2010 *IRILL* [www.irill.org](http://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

- 1 Introduction
- 2 Software and Open Science
- 3 An emerging policy framework
- 4 Assessing the needs and a strategy to address them
- 5 Meet Software Heritage (and the HAL connection)
- 6 Demo time!
- 7 Software Heritage, cont'd
- 8 Actions





# Why Open Science?

Open Science ([Second National Plan for Open Science](#), France, 2021)

*Unhindered* dissemination of results, methods and products from scientific research. It draws on *the opportunity provided by recent digital progress* to develop *open access* to *publications* and – as much as possible – *data, source code and research methods*.

Jean-Eric Paquet (EU DGRI, [on the objective of Open Science](#))

“Increase *scientific quality*, the *pace of discovery and technological development*, as well as *societal trust in science*.”

Mariya Gabriel ([EU Commissioner](#) for Research)

The COVID-19 crisis has also shown that cooperation at international level in research and innovation is more important than ever, including through *open access to data and results*. *No nation, no country can tackle any of these global challenges alone*.

Yuval Noah Harari (on COVID 19)

“*The real antidote [to epidemic] is scientific knowledge and global cooperation*.”

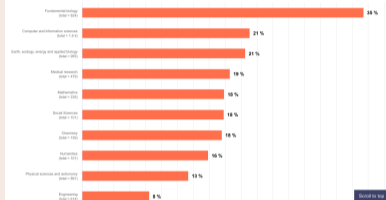
# Software is a pillar of Open Science

## Software powers modern research

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

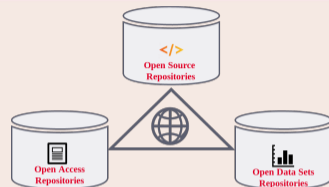
Sort by:

Highest volume  Highest sharing rate



Over 20% of articles across all disciplines share software  
2023 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.”*

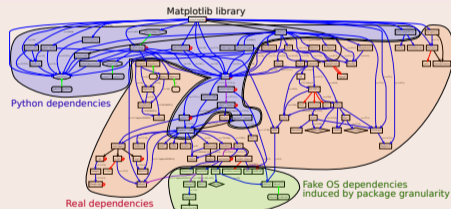
# Software source code is *not* data (and FAIR is not the answer)

## 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 ...

# 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

awareness is raising at the level of public policy

- 1 Introduction
- 2 Software and Open Science
- 3 An emerging policy framework**
- 4 Assessing the needs and a strategy to address them
- 5 Meet Software Heritage (and the HAL connection)
- 6 Demo time!
- 7 Software Heritage, cont'd
- 8 Actions



# International highlights

## Paris Call on Software Source code (2019, UNESCO)

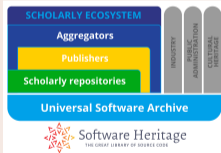


40 international experts call to “promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, [...] recognising in the careers of academics their contributions to high quality software development, in all their forms”



Open Source in UNESCO [recommendations](#) for Open Science, 2021

## Software in the EOSC



2020 [EOSC SIRS](#) connect scholarly ecosystem via Software Heritage

2021 [EOSC Task Force](#) on Infrastructures for Research Software

2022 [FAIRCORE4EOSC project](#) WP6 implements SIRS report

2023 [INFRAEOSC call](#) on quality of scientific software

## 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



1

## Second French Plan for Open Science



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 licence 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

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

## Leveraging experience and connections

- Open Source thematic group in Systematic (since 2007, more on demand)
- Collaboration with DINUM, Eclipse Foundation, OW2, ...

## 20+ active members

### Chairs: Roberto Di Cosmo and François Pellegrini

- Florent CHUFFART (Univ Grenoble Alpe)
- Mélanie CLÉMENT-FONTAINE (Univ Paris-Saclay - Versailles Saint-Quentin)
- Laurent COSTA (UMR 7041 ArScAn)
- Ludovic COURTÈS (Inria)
- Sébastien GÉRARD (Univ Paris-Saclay, CEA, List)
- Mathieu GIRAUD (CNRS, Univ Lille)
- Timothée GIRAUD (CNRS)
- Jean-Yves JEANNAS (Univ Lille, AFUL)
- Nicolas JULLIEN (IMT Atlantique)
- Daniel LE BERRE (Univ Artois, CNRS)
- Violaine LOUVET (CNRS / GRICAD - Univ Grenoble Alpes)
- Camille MAUMET (Inria, Univ Rennes, CNRS, Inserm)
- Clémentine MAURICE (CNRS)
- Grégory MIURA (Univ Bordeaux Montaigne)
- Raphaël MONAT (LIP6, Sorbonne Université)
- Patrick MOREAU (CNRS)
- Sophie RENAUDIN (AP-HP)
- Nicolas ROUGIER (Inria, Univ Bordeaux, CNRS)
- François SABOT (IRD)
- Sylvie TONDA-GOLDSTEIN (Inria)
- Samuel THIBAUT (Univ Bordeaux) (Univ Paris-Saclay)

- 1 Introduction
- 2 Software and Open Science
- 3 An emerging policy framework
- 4 Assessing the needs and a strategy to address them
- 5 Meet Software Heritage (and the HAL connection)
- 6 Demo time!
- 7 Software Heritage, cont'd
- 8 Actions



## 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.)

Here we will focus on ARDC

# 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

Artifacts Available      Artifacts Evaluated & Functional

Authors/Contributors: [Authors Info & Affiliations](#)

DOI: <https://doi.org/10.1145/> [redacted] Version: 1.0

**Description**

A source archive of [redacted], and the version of [redacted] used in the paper eval. A more up-to-date version of [redacted] can be found at [github.com/\[redacted\]/\[redacted\].git](https://github.com/[redacted]/[redacted].git)

**Assets**

Read Me [redacted]

[Download \(3.5 KB\)](#)

## 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 Software and Open Science
- 3 An emerging policy framework
- 4 Assessing the needs and a strategy to address them
- 5 Meet Software Heritage (and the HAL connection)**
- 6 Demo time!
- 7 Software Heritage, cont'd
- 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



United Nations  
Educational, Scientific and  
Cultural Organization



And many more ...

[www.softwareheritage.org/support/testimonials](http://www.softwareheritage.org/support/testimonials)

## Donors, members, sponsors



Diamond sponsor



Platinum sponsors



Gold sponsors



Silver sponsors



Bronze sponsors



# The largest software archive, a shared infrastructure

Cultural Heritage



Industry



Research



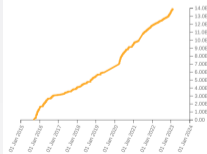
Public Administration



## Software Heritage

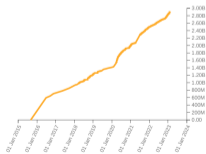
Source files

13,974,813,954



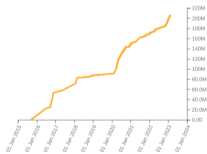
Commits

2,912,845,019



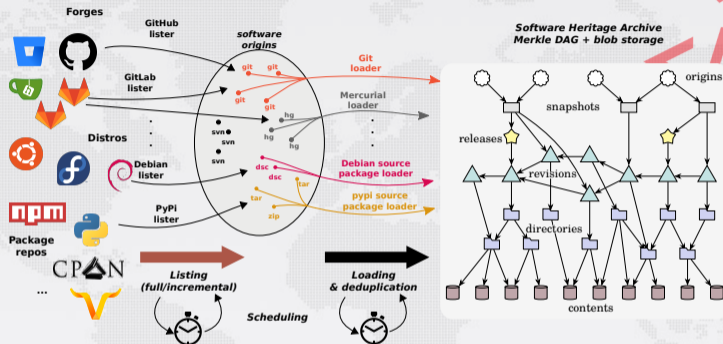
Projects

207,160,527



<b>Bitbucket</b> 1,925,997 origins	<b>git</b> 21,603 origins	<b>R</b> 21,113 origins
<b>debian</b> 128,719 origins	<b>Go</b> 5,947 origins	<b>GitHub</b> 137,564,899 origins
<b>GitLab</b> 3,982,586 origins	<b>Guix</b> 12,032 origins	<b>GNU</b> 354 origins
<b>heptapod</b> 1,068 origins	<b>launchpad</b> 329,908 origins	<b>Maven</b> 93,738 origins
<b>NixOS</b> 12,032 origins	<b>npm</b> 1,799,296 origins	<b>Cargo</b> 4,083 origins
<b>Phabricator</b> 192 origins	<b>python</b> 410,582 origins	<b>SOURCEFORGE</b> 308,990 origins

# Address common Open Science and Open Source needs: archival



Global development history permanently archived in a uniform data model

- over 14 billion unique source files from over 210 million software projects
- ~1PB (compressed) blobs, ~30 B nodes, ~400 B edges

# A peek under the hood: growing set of listers and loaders

## Supported listers (index)

Software Heritage - User Documentation

Software Heritage  
THE GREAT LIBRARY OF SOURCE CODE

Search docs

CONTENTS:  
Frequently Asked Questions

Software Heritage listers

- Arch lister
- AUR lister
- Bitbucket lister
- Bower lister
- Cgit lister
- CPAN lister
- CRAN lister
- Crates lister
- Debian lister
- Gitea lister
- GitHub lister
- GitLab lister





Software Heritage listers

View page source

### Software Heritage listers

A **lister** is a software component used for the discovering of software origins to load into the Software Heritage archive.

This page references all available listers and links to their high-level documentation.

Lister name	Related links	Current status	Related grants
 Arch lister	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 AUR lister	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 Bitbucket lister	<ul style="list-style-type: none"><li>Source code</li><li>Developer doc</li><li>Development</li></ul>	in production	
 Bower lister	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	NLNet Foundation (awarded to Octobus)

## Supported loaders (index)

Software Heritage  
THE GREAT LIBRARY OF SOURCE CODE

Search docs

CONTENTS:  
Frequently Asked Questions  
Software Heritage listers

Software Heritage loaders






- Arch loader
- Archive loader
- AUR loader
- Bazaar loader
- CRAN loader
- Crates loader
- CVS loader
- Debian loader
- Deposit loader
- Git loader
- Golang loader
- Hackage loader
- Maven loader
- Mercurial loader
- Nix/Guix loader
- NPM loader

Software Heritage loaders

View page source

A **loader** is a software component used to ingest content into the Software Heritage archive.

This page references all available loaders and links to their high-level documentation.

Loader name	Related links	Current status	Related grants
 Arch loader	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 Archive loader	<ul style="list-style-type: none"><li>Source code</li><li>Developer doc</li></ul>	in production	
 AUR loader	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 Bazaar loader	<ul style="list-style-type: none"><li>Source code</li><li>Developer doc</li><li>Development</li></ul>	in production	Alfred P. Sloan Foundation (awarded to Octobus)
 Maven loader	<ul style="list-style-type: none"><li>Source code</li></ul>		

Many contributed from external experts

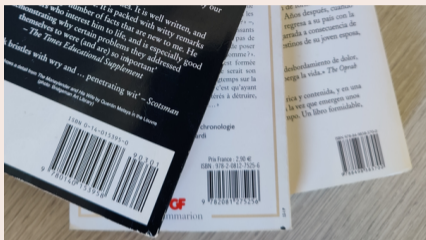
thanks to support of Alfred P. Sloan and NLNet foundations

## Software Heritage is a game changer

- **universal** archive for **all** the source code
- uniform **intrinsic** identifier, independent of the development platform or technology

Let's revisit what we really mean by *referencing* a source code

## Identification of a book



Goal: identify a book

- one ISBN number per published book
- ISO 2108 standard specification

## Location of (a copy of) a book



Goal: find (a copy of) a book

- many locations (locations can change!)
- many approaches for call numbers

identification and location are *separate concerns*

# 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.

	<b>Extrinsic</b>	<b>Intrinsic</b>
relation	register	convention
persistence	external <sup>a</sup>	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>

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

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

we need a

# Meet the SWHID identifier

## Software Heritage Identifiers (SWHID)

[link to full docs](#)



25+B  
intrinsic,  
decentralised,  
cryptographic

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

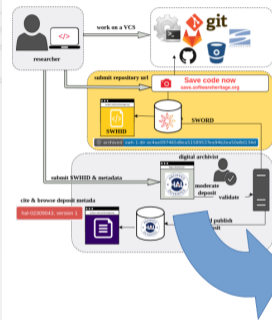
- Linux Foundation [SPDX 2.2](#)
- IANA-registered "swh:"
- WikiData property [P6138](#)

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

**Breaking news: standardisation**, see [swhid.org](#)



# HAL and Software Heritage: building a curated software catalog



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

The screenshot shows the HAL website interface for the **LinBox** repository. The page title is **LinBox** and the version is **1.0.0.A.0.1.0.0**. The repository is associated with the **CC-BY** license. The page includes a list of authors and affiliations, such as **INRIA - Arithmétique et Calcul**, **INRIA Grenoble - Rhône-Alpes, LIP - Laboratoire de l'Informatique du Parallélisme**, and **AVAILON - Algorithms and Software Architectures for Distributed and HPC Platforms**. The page also features a **Metadata** section with fields for **Version**, **Software License**, **Programming Language**, and **Code Repository**. A **COLLECTIONS** section lists various categories like **ENS LYON | CNRS | LIRMM | ECC | UNIV-LYON1 | MPS | INRIA | UNIV-SANTOPIERRE**. The **DESCRIPTION** section provides a detailed overview of the LinBox library, including its purpose and the types of algorithms it supports. The **EXPORT** section offers options to download the metadata in **CSV**, **Bibtex**, **JSON**, **DC**, or **Other** formats.

The screenshot shows the **Browse the archive** interface for the **config-bias.h** file. The page displays the file's metadata, including the **Revision** **e818328952266b785c62963b11963b1496107** and the **Programming Language** **C++**. The file's content is shown as a code block, starting with a comment: `/* config-bias.h`. The code includes copyright information for **Pascal Giorgi** and **2007 Clement Fernet**, and a license notice: `* This file is part of the Library LinBox.` The code also includes a `#ifndef LINBOX_CONFIG_BIAS_H` guard. The page also features a **Tip revision** section and a **Permissions** sidebar.

[swh:1:dir:393b611a1424f032e83569bf6762502371cfc65](https://www.softwareheritage.org/doi/10.26434/chemrxiv-2019-08-12-utc)

with minimal user overhead!

- 1 Introduction
- 2 Software and Open Science
- 3 An emerging policy framework
- 4 Assessing the needs and a strategy to address them
- 5 Meet Software Heritage (and the HAL connection)
- 6 Demo time!
- 7 Software Heritage, cont'd
- 8 Actions



# A walkthrough

- Browse (e.g. [Apollo 11](#), and 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](#)

# A look at some adoption indicators

## From [Melissa Harrison's OSEC 2022 talk](#)

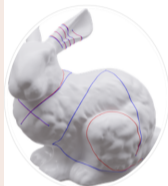


### What are they "referencing"?

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"

## Upcoming on [replicabilitystamp.org](https://replicabilitystamp.org) (please do not spoil :-))



### b/Surf: Interactive Bézier Splines on Surface Meshes

Claudio Mancinelli, Giacomo Nazzaro, Fabio Pellacini, Enrico Puppo  
IEEE Transactions on Visualization and Computer Graphics (TVCG)



 Repository



## HAL+SWH in the Open Science software booklet

## Funding agencies recommendations [ANR 2023 guidelines](#) (p. 17)

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

- 1 Introduction
- 2 Software and Open Science
- 3 An emerging policy framework
- 4 Assessing the needs and a strategy to address them
- 5 Meet Software Heritage (and the HAL connection)
- 6 Demo time!
- 7 Software Heritage, cont'd**
- 8 Actions





## National Roadmap of Research Infrastructures (2022-...)

### Framework

*"The strategic, scientific and financial challenges related to research infrastructures are such that sooner or later we will need to jointly program European and national investments in major research facilities."*

Frédérique Vidal, Minister of Research, 2018

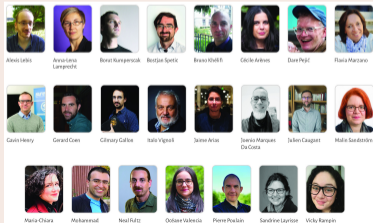
Software Heritage is now listed

# A growing and active community

## Team



## Ambassadors



R. Di Cosmo    roberto@dicosmo.org    (CC-BY 4.0)

## Contributors to the platform

```
13:21:32 <seirf> from last time i ran it? it very likely is
13:21:47 <seirf> we had a x2 on the edges in a single year
13:23:14 <vlorentz> ah
13:53:44 <zack> seirf: i think i was remembering the LLP time on granet rather than the one (on the previous
graph) on the big mem telecom machine
13:54:01 <zack> wasn't it something like 10-14 days (on granet)?
13:55:11 <seirf> zack: it depends on the number of weights you use
13:55:23 <seirf> i had something like that to do the parameter sweep
13:55:31 <seirf> but then i settled on a few good gamma values
13:55:44 <seirf> and afterwards it was only ever ~3-4 days
14:02:57 <zack> ok
15:19:35 <jelmer> vlorentz: when is jenkins meant to kick in ? I didn't think the CI would mean you pasting test
results in comments :P
15:19:59 <jelmer> alternatively, i could try to get it working locally - for some reason tox doesn't run here,
complaining it can't find swh.scheduler
15:20:48 <vlorentz> jenkins is down until tomorrow evening (paris time)
15:20:59 <vlorentz> bad day for submitting your code :D
15:21:18 <vlorentz> er yeah i just fixed that issue
15:21:31 <vlorentz> but the fixed swh.scheduler is not pushed to pypi because Jenkins
15:23:25 <jelmer> ah
15:23:40 <vlorentz> in the meantime, you can change apply this patch: https://gitlab.softwareheritage.org/-/snippets/1546
15:23:44 <vlorentz> as an ugly workaround
15:24:13 <vlorentz> actually, just adding "pytest-postgresql < 4.0.0" should do it
15:25:00 <vlorentz> when jenkins is back online i'll push a new release of swh-scheduler without the missing
dependency on pytest-postgresql
```

- Nicks
- Alphare
  - ames
  - amj
  - amoeba
  - ar-jan
  - bchauvet[m]
  - cmatrix[m]
  - dan-
  - deesix
  - ericson2314
  - francbret1
  - guerby
  - Guest92
  - haltode2
  - hpliar
  - jayeshiv
  - jelmer
  - KSHivendu
  - landre[m]
  - marmoute

## Awards

★ **Antoine Pietri**, best French PhD in Software Engineering "Enabling Big Code analysis on a very large source code corpus". Awarded by the CNRS research working group GPL <https://theses.hal.science/tel-03515795v1>

★ **Stefano Zacchiroli with Davide Rossi**, Google Award for Inclusion Research 2022, for the research project "What Causes the Lack of Diversity in Open Source?". <https://research.google/outreach/air-program/recipients/>

★ **Antoine Pietri with Stefano Zacchiroli**, Award Best Dataset Paper: "A Large-scale Dataset of (Open Source) License Text Variants". <https://arxiv.org/abs/2204.00256>

## Annual report



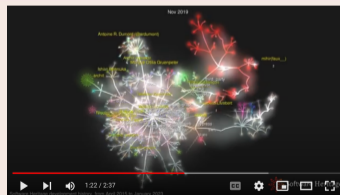
## 5 years in 5 minutes

[Link](#)



## Evolution of our codebase

[Link](#)





- 1 Introduction
- 2 Software and Open Science
- 3 An emerging policy framework
- 4 Assessing the needs and a strategy to address them
- 5 Meet Software Heritage (and the HAL connection)
- 6 Demo time!
- 7 Software Heritage, cont'd
- 8 **Actions**



# Call to action: best practices for ARDC are available... today!

## Archiving and referencing

For **all source code** used in research (*yes, even small scripts!*)

- ensure it is archived in Software Heritage (see [save code now](#))
- get the proper **SWHID** for your software (see [detailed HOWTO](#))
- add it to research articles for reproducibility (see [detailed HOWTO](#))

## Describing and Citing/Crediting

For **software you want to put forward** (*mention in your CV, reports, etc., get citations and credit for it*), do the following **extra steps**:

- add **codemeta.json** with description (see the [codemeta generator](#))
- reference in the HAL portal (french partners, see [online HAL documentation](#))
- cite software using the [biblatex-software](#) package (in CTAN and TeXLive)

- train students and colleagues
- engage journals, conferences, learned societies






## A working agenda

- avoid proprietarisation: set the default to open
  - *publicly funded research software should be open source, exceptions must be justified*
- avoid balkanisation
  - build on common, shared, open, non profit infrastructures, like Software Heritage
- support mutualised common infrastructures
  - acknowledge the **predominant human component** of digital infrastructures
    - recurrent funding of their cost
    - proper evaluation of their service
- remember *Goodhart's Law*:  
*when a measure becomes a target, it stops being a good measure*
  - establish intelligent incentives
  - count quality software contributions in careers, avoid purely numerical indicators, keep the human in the loop

it's a long road, but together we can make it

## Questions?

### 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](#))