Vers un pilier logiciel de la Science Ouverte

Roberto Di Cosmo

Director, Software Heritage
Inria and Université de Paris Cité

13 Décembre 2022

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline

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

2010 *IRILL* www.irill.org

2015 *Software Heritage* at INRIA

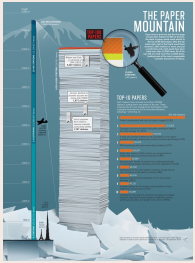
2018 *National Committee for Open Science*, France

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

Outline

Software is a pillar of Open Science

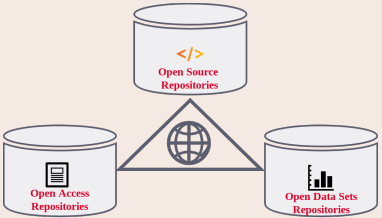
Software powers modern research



[...] software [...] essential in their fields.
Top 100 papers (Nature, 2014)

Sometimes, if you dont have the software, you dont have the data
Christine Borgman, Paris, 2018

A key pillar: software (source code)



The links in the picture 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)

```
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
```

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."

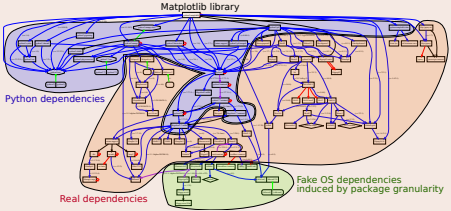
Software source code is *not* data (and FAIR is not the silver bullet!)

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*



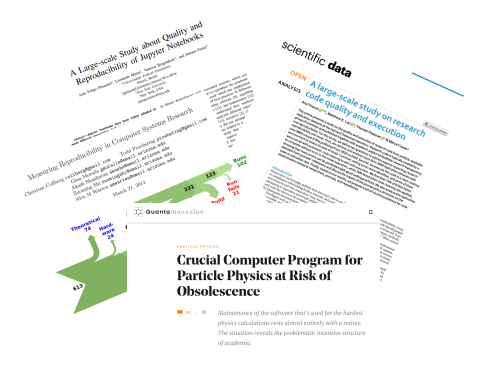
The human side

design, algorithm, code, test, documentation, community, funding

and so many more facets …

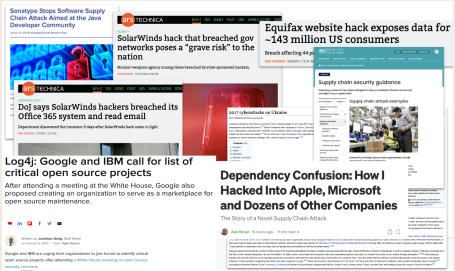
How are we managing our software ?

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

Outline

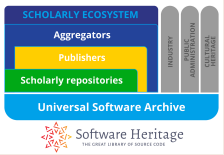
International highlights

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

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

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

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

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

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 **reproductibility** of scientific results. »

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

3

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

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

Software College in the CoSO

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, …

Software College in the CoSO, cont'd

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 THIBAULT (Univ Bordeaux) (Univ Paris-Saclay)

Outline

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.)

Here we will focus on ARDC

A word of caution before moving on

Work *with* researchers and engineers

- recommendations need *appropriate and actionable support*
- adoption is easier if one *provides value* for researchers

In particular

- avoid unnecessary overhead
- ask only once
- adapt requirements to the maturity level

Some popular approaches, and why they do not fit the bill

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

.zip or .tar file on:

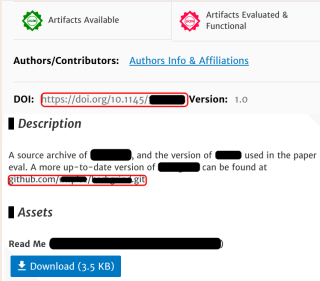
- ~~ftp server~~
- web page
- document archive (+ DOI)

B - Since the 2000's

Rely on *software forges*

- institutional or project ones
- free commercial ones: BitBucket, GitHub, GitLab, …

C: a mix of the two



Can get no satisfaction…

A *Poor user experience*

B *Preservation?*

C Can do 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

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

An international, non profit initiative built for the long term

Sharing the vision

United Nations
Educational, Scientific and
Cultural Organization

And many more ...
www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Diamond sponsor

Platinum sponsors

Gold sponsors

Silver sponsors

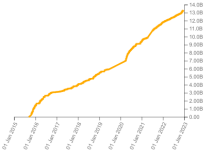
Bronze sponsors

The largest software archive, a shared infrastructure

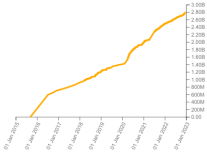
Cultural Heritage | Industry | Research | Public Administration

Software Heritage

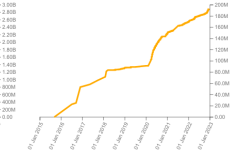
Source files
13,338,879,609



Commits
2,801,474,518



Projects
193,279,669



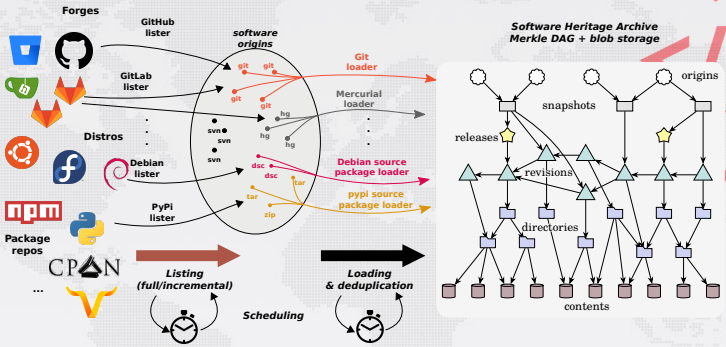
Directories
10,905,976,959

Authors
51,691,686

Releases
35,938,300

Bitbucket 1,925,997 origins	git 21,603 origins	R 21,113 origins
debian 128,719 origins	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	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 12 billion unique source files from over 180 million software projects
- ~1PB (uncompressed) blobs, ~25 B nodes, ~350 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	• Source code • Development	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
AUR lister	• Source code • Development	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
Bitbucket lister	• Source code • Developer doc • Development	in production	
Bower lister	• Source code • Development	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
- NixGuix loader
- NPM loader

» Software Heritage loaders — View page source

Software Heritage loaders

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	• Source code • Development	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
Archive loader	• Source code • Developer doc	in production	
AUR loader	• Source code • Development	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
Bazaar loader	• Source code • Developer doc • Development	in production	Alfred P. Sloan Foundation (awarded to Octobus)
	• Source code		

Many contributed from external experts

thanks to support of Alfred P. Sloan and NLNet foundations

Address common Open Science and Open Source needs: reference

Software Heritage Identifiers (SWHID)

link to full docs

25+B
intrinsic,
decentralised,
cryptographic

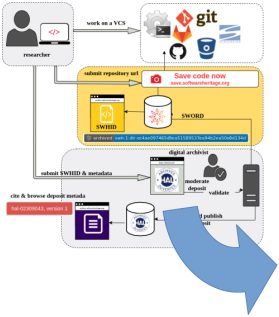
Full fledged *source code references* for traceability, integrity and reproducibility

- Linux Foundation SPDX 2.2
- IANA-registered `"swh:"`
- WikiData property P6138

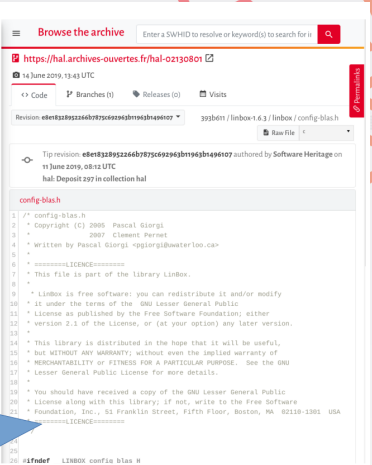
Examples: Apollo 11 AGC excerpt, Quake III rsqrt
Guidelines available, see the HOWTO

Breaking news: standardisation, see swhid.org

HAL and Software Heritage: building a curated software catalog



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



swh:1:dir:393b611a1424f032e83569bf6762502371cfcf65

with minimal user overhead!

Outline

A walkthrough

- Browse (e.g. Apollo 11, and your work may be already there !)
- Trigger archival, use the updateswh browser extension (GitHub action available too)
- 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

Outline

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

Call to action: policy making

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

The floor is yours

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)