

# Software Heritage — Building the Software Pillar of Open Science

Stefano Zacchiroli

Télécom Paris  
Institut Polytechnique de Paris



# Software is everywhere



# Software Source Code



*“The source code for a work means the preferred form of the work for making modifications to it.”*

GPL License

Hello World

Program (excerpt of binary)	Program (source code)
4004e6: 55 4004e7: 48 89 e5 4004ea: bf 84 05 40 00 4004ef: b8 00 00 00 00 4004f4: e8 c7 fe ff ff 4004f9: 90 4004fa: 5d 4004fb: c3	/* Hello World program */ <pre>#include&lt;stdio.h&gt; void main() {     printf("Hello World"); }</pre>

# 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    GOTOPOOH   # 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 threehalves = 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 * ( threehalves - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalves - ( 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.”*

# A lightning fast growth

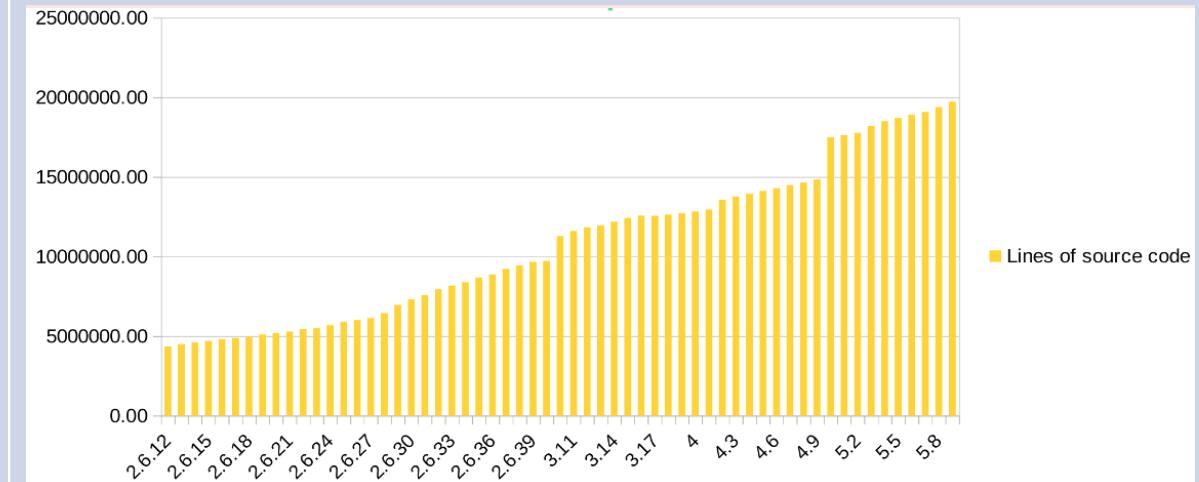
Apollo 11 (~60.000 lines), 1969



"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton

Linux Kernel : 20 million lines. . .



. . . now in your pockets!

Open source software is eating the software world

tens of millions of developers collaborate on open source software worldwide today

Reuse is the new rule

80% to 90% of a new application is... just reuse!

(Sonatype survey, 2017)

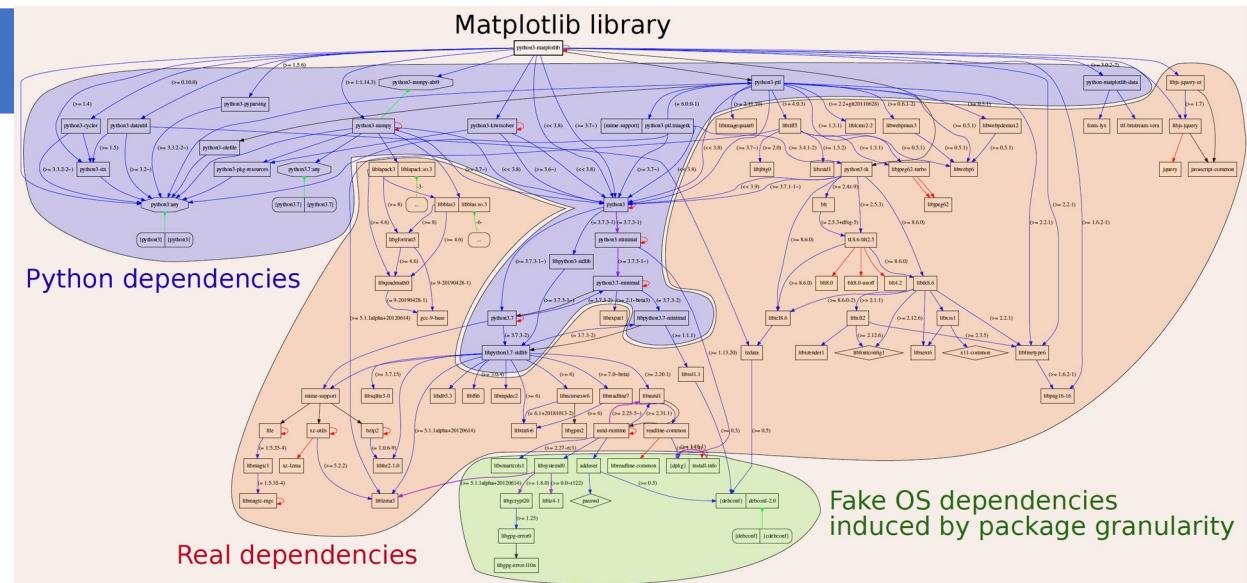
# Source code is *special*: software is *not* data

# 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*



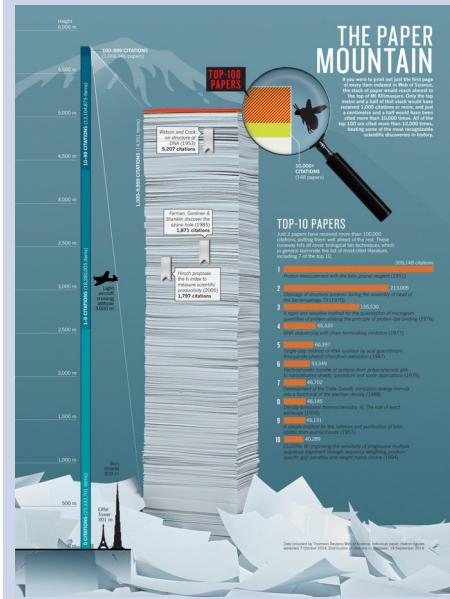
## The human side

copyright law applies!

fruit of human ingenuity: design, algorithm, code, test, documentation, community, funding, and so many more facets...

# A long overlooked pillar of Open Science

# Software powers modern research



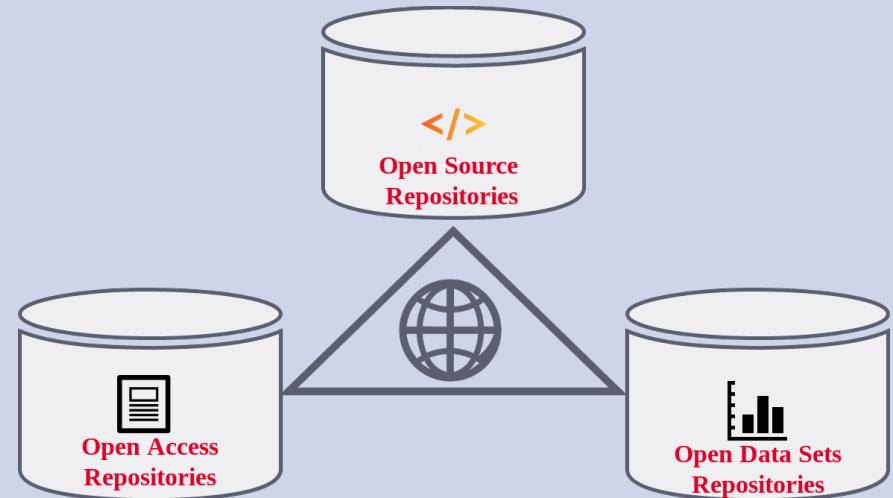
[...] software [...] essential in their fields.

## *Top 100 papers (Nature, 2014)*

*Sometimes, if you don't have  
the software, you don't have the  
data*

*Christine Borgman, Paris, 2018*

# Missing pillar: software (source code)



The links in the picture are essential

## Software may be a **tool**, an **outcome** and a **research object**

Open source (open access to the source code) is necessary

- avoid reinventing the wheel, accelerate scientific discoveries
  - **preserving source code** and its history is necessary for *reproducibility*

# A plurality of needs

## Researchers

- O archive and reference software used in articles
- O find useful software
- O get credit for software contributions
- O verify, reproduce, improve results

## Laboratories/teams

- O track software contributions
- O produce reports
- O maintain webpage

## Research Organization

- know its software assets
- O technology transfer
- O impact metrics
- O funding strategy
- O career evaluation

# What is at stake: ARDC

in increasing order of difficulty

## Archive

Research software artifacts must be properly **archived**

make sure we can *retrieve* them (*reproducibility*)

## Reference

Research software artifacts must be properly **referenced**

make sure we can *identify* them (*reproducibility*)

## Describe

Research software artifacts must be properly **described**

make it easy to *discover* and reuse them (*visibility*)

## Cite/Credit

Research software artifacts must be properly **cited** (*not the same as referenced!*)

to give credit to authors (*evaluation!*)

# Software Heritage in a nutshell

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



**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

damage  
disaster  
media  
malicious  
aging  
attack  
obsolete  
dependencies  
dangling  
reference  
deletion  
storage  
wear  
corruption  
encryption  
format

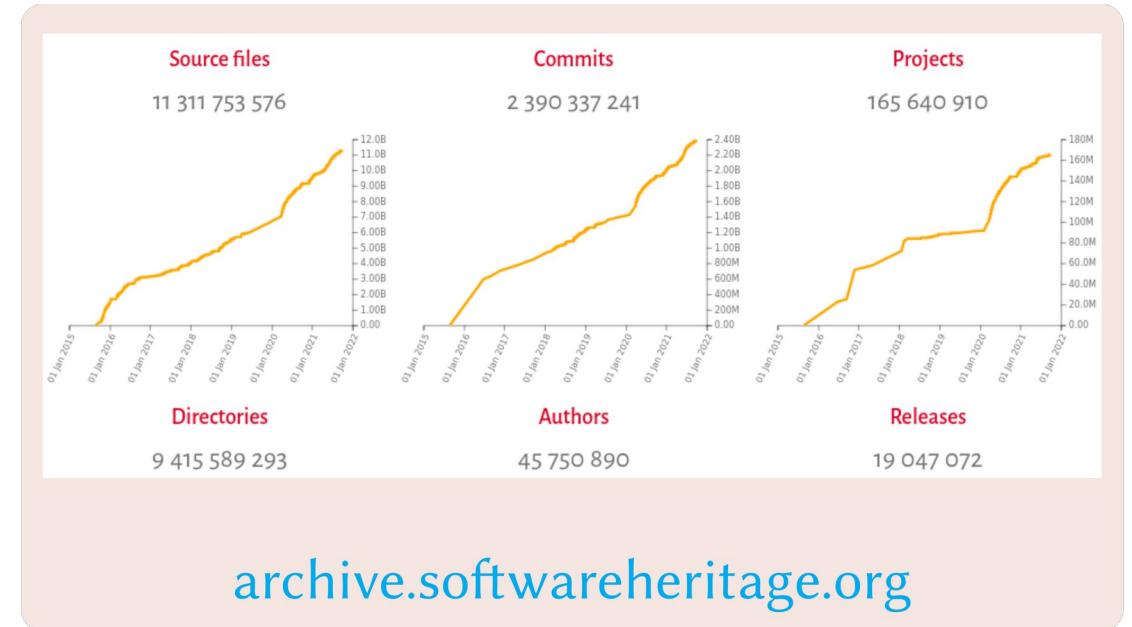
**preserve** all software  
source code

## Research infrastructure



**enable analysis** of all  
software source code

# The largest public source code archive, principled



## Technology

- transparency and FOSS
- replicas all the way down

## Content (billions!)

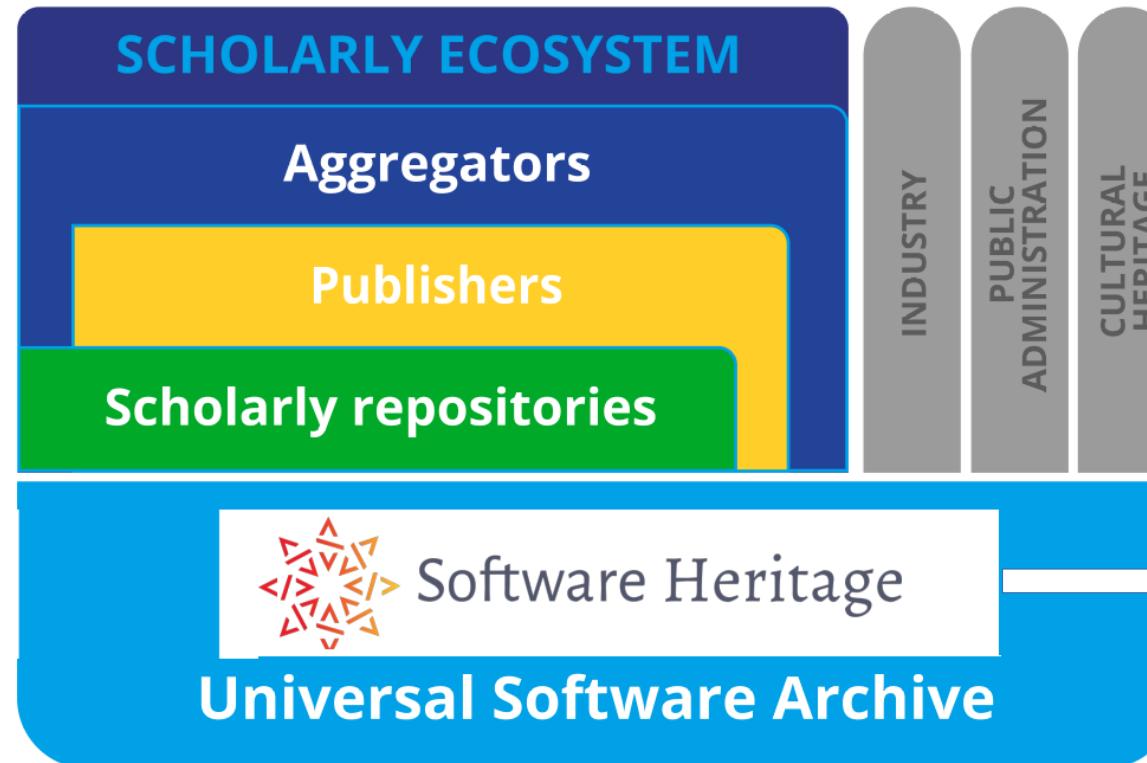
- intrinsic identifiers
- facts and provenance

## Organization

- non-profit
- multi-stakeholder

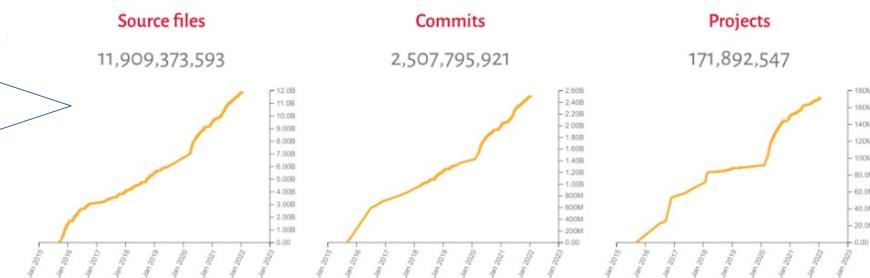
# The big picture (EOSC SIRS 2020 report)

## Research Software Infrastructures: Overall Architecture



### Scholarly ecosystem

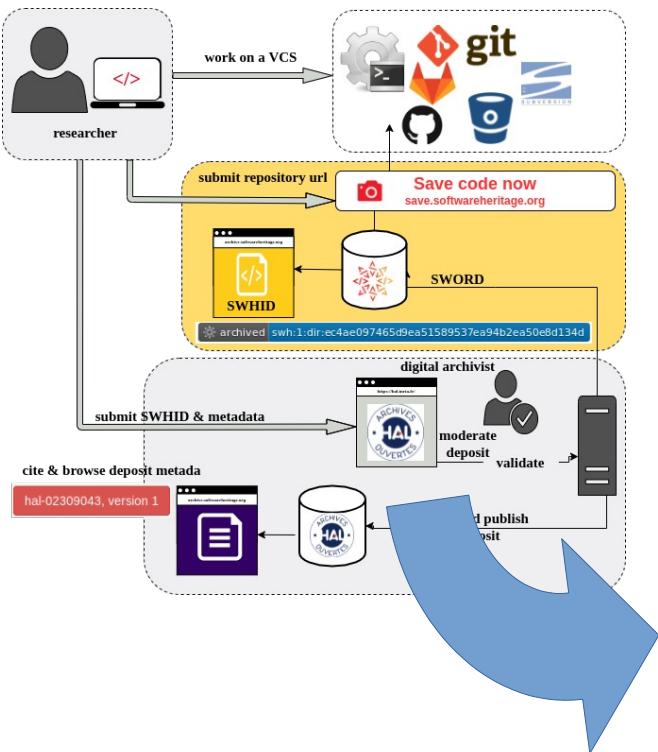
- Aggregators collecting data from...
- Academic publishers
- Scholarly repositories



### Key observations

- One common layer for **archive and reference** shared with **all the software world**
- Added value in the scholarly ecosystem: **curation, citation and credit**

# The HAL – Software Heritage success story



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

LinBox  
The LinBox Group 1, 2, 3, 4, 5, 6, 7, 8, 9 [Details]  
1 ECO - Exact Computing  
LIRMM - Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier  
2 ARIC - Arithmetic and Computing  
Inria Grenoble - Rhône-Alpes, LIP - Laboratoire de l'Informatique du Parallelisme  
3 AVALON - Algorithms and Software Architectures for Distributed and HPC Platforms  
Inria Grenoble - Rhône-Alpes, LIP - Laboratoire de l'Informatique du Parallelisme  
4 CIS - Department of Computer and Information Sciences [Newark]  
5 Drexel University  
6 NCSU - Department of Mathematics [Raleigh]  
7 United States Naval Academy  
8 SCG - Symbolic Computation Group  
9 CASC - Calcul Algébrique et Symbolique, Sécurité, Systèmes Complexes, Codes et Cryptologie  
LJK - Laboratoire Jean Kuntzmann

Abstract : LinBox is a C++ template library of routines for solution of linear algebra problems including linear system solution, rank, determinant, minimal polynomial, characteristic polynomial, and Smith normal form. Algorithms are provided for matrices with integer entries or entries in a finite field. A number of matrix storage types is provided, especially for blackbox representation of sparse or structured matrix classes. A few algorithms for rational matrices are available. LinBox also uses underlying data structures and algorithms for integer, rational, polynomial, finite fields and rings, as well as dense and sparse matrix formats coming from the Giavro (<https://casy.sicard.pages.univ-grenoble-alpes.fr/giavro/>) and FFLAS-FFPACK (<http://linbox-team.github.io/fflas-ffpack/>) libraries.

Document type : Software  
Domain : Computer Science [cs]  
Computer Science [cs] / Symbolic Computation [cs.SC]

BROWSE

swh:1:dir:393b611a1424f032e83569bf6762502371cf65:origin=http://hal.archives-ouvertes.fr/hal-02130801:visit=swh:1:snp:19c29b988fe02623c70c7dc8b97c42481eb691b:anchor=swh:1:rev:e8e18328952266b7875c692963b11963b1496107:path=/ (hal-02130801)

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

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

14 June 2019, 13:43 UTC

Code Branches (1) Releases (0) Visits

Revision: e8e18328952266b7875c692963b11963b1496107 393b611 / linbox-1.6.3 / linbox / config-blas.h

Raw File

Tip revision: e8e18328952266b7875c692963b11963b1496107 authored by Software Heritage on 11 June 2019, 08:12 UTC

hal: Deposit 297 in collection hal

config-blas.h

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

#ifndef LINBOX config blas H
```

swh:1:dir:393b611a1424f032e83569bf6762502371cf65

# What is needed

## Policy for dissemination and reuse

- Set the default to open source for research software
- Open source creates value : adapt technology transfer and industry collaboration to it

## Framework for evaluation and recognition

- Make software development count in a career (not the case in many countries)
- Avoid purely quantitative indicators

## Sustainability

- |                       |  |
|-----------------------|--|
| <b>Technical</b>      | improve quality of key research software                               |
| <b>Organisational</b> | professional practices for governance and maintenance                  |
| <b>Financial</b>      | make open source research software as easy to fund as buying a license |

# Good news: awareness is raising

## Paris Call on Software Source code (2019)

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

<https://en.unesco.org/foss/paris-call-software-source-code>

## EOSC SIRS report (2020) and EOSC TF on infrastructures for research software

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

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

## UNESCO Open Science Recommendations (2021)

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

# Focus on the 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



### 2nd National Plan for Open Science (6/7/2021)

#### Open and promote research software source code

##### O actions (selection)

- O charter for research software policy
- O recognize software development (see [the 2021 prize](#))
- O coordinate communities of practice
- O build a connected ecosystem of research outputs

##### O recommendations (selection)

- O **Archive source code in Software Heritage**
- O **Standardize and use SWHID**
- O **Build a national catalog of research software**

See [official announcement](#)

Let's build together the software pillar of open science  
it's a long road, but together we can make it

## Questions

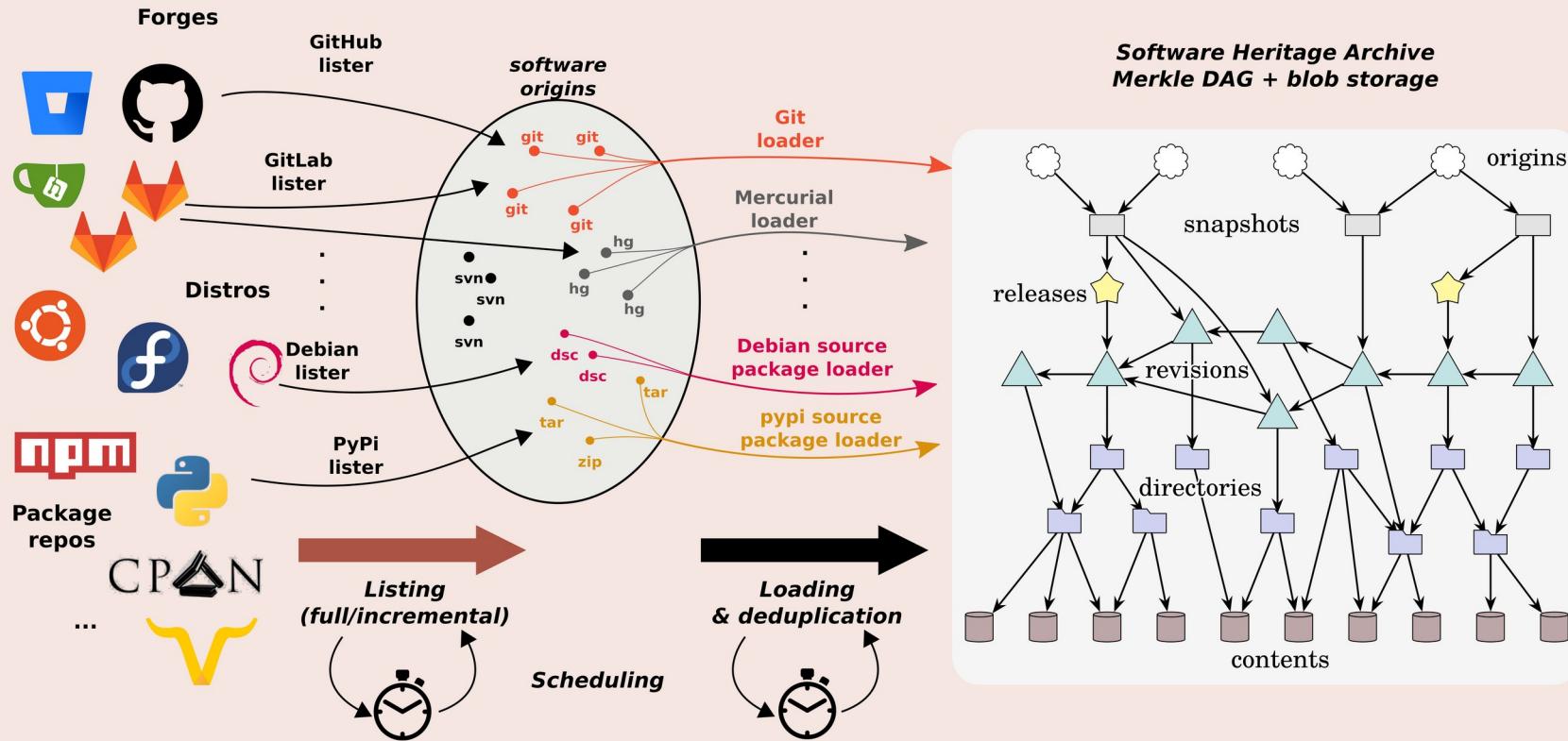
### References

- 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.1145/3183558))

Slide credits: © R. Di Cosmo, S. Zacchiroli CC-BY 4.0

# Appendix

# A peek under the hood



A **global graph** linking together fully **deduplicated** source code artifact (files, commits, directories, releases, etc.) to the places that distribute them (e.g., Git repositories), providing a **unified view** on the entire *Software Commons*.  
(Size: ~20 B nodes, ~200 B edges, ~900 TB blobs)

# Software Heritage Identifiers (SWHIDs)



An emerging standard

- in Linux Foundation's [SPDX 2.2](#)
- IANA registered, WikiData property [P6138](#)

Examples:

- [Apollo 11 AGC excerpt](#)
- [Quake III rsqrt](#)

# An international non-profit initiative

## 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

*Inria*

Diamond sponsor



Platinum sponsors



Gold sponsors



Silver sponsors



Bronze sponsors

