Towards a Software Pillar for Open Science

leveraging the universal source code archive

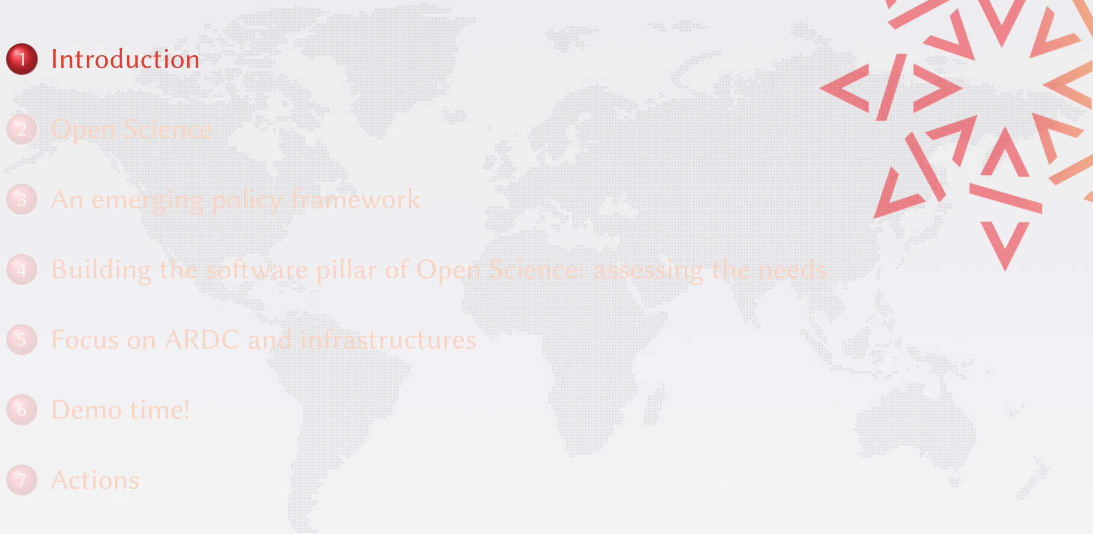
Roberto Di Cosmo

Director, Software Heritage
Inria and Université de Paris Cité

June 2022

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline



1. Introduction
2. Open Science
3. An emerging policy framework
4. Building the software pillar of Open Science: assessing the needs
5. Focus on ARDC and infrastructures
6. Demo time!
7. 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
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

1. Introduction
2. Open Science
3. An emerging policy framework
4. Building the software pillar of Open Science: assessing the needs
5. Focus on ARDC and infrastructures
6. Demo time!
7. 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 Commissionneer for Research)

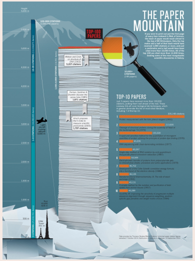
The COVID-19 crisis has also shown that cooperation at international level in research and innovation is more important than ever, including through *open access to data and results. No nation, no country can tackle any of these global challenges alone.*

Yuval Noah Harari (on COVID 19)

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

Software is a pillar of Open Science

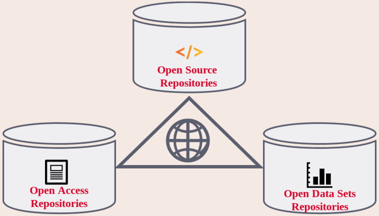
Software powers modern research



[...] 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!

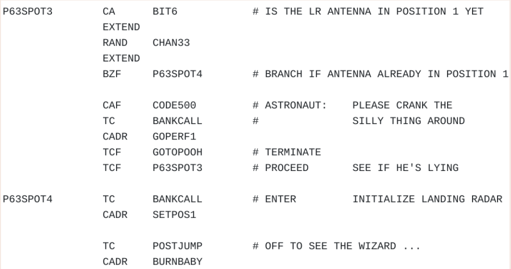
Preserving (the history of) source code is necessary for *reproducibility*

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)



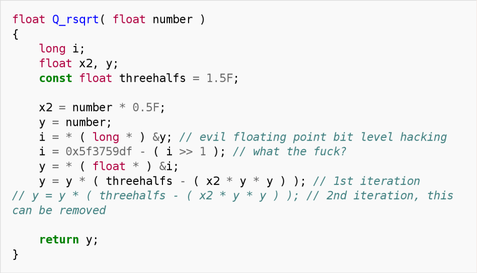
```
P63SPOT3        CA      BIT6            # IS THE LR ANTENNA IN POSITION 1 YET
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     P63SPOT4        # BRANCH IF ANTENNA ALREADY IN POSITION 1

                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.”

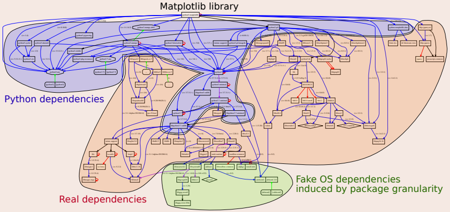
Source code is *special* (software is *not* data)

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

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

and so many more facets …

Outline

1. Introduction
2. Open Science
3. An emerging policy framework
4. Building the software pillar of Open Science: assessing the needs
5. Focus on ARDC and infrastructures
6. Demo time!
7. Actions

International highlights

Paris Call on Software Source code (2019, UNESCO)

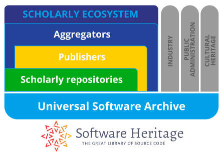
40 international experts call to *“promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, [...] recognising in the careers of academics their contributions to high quality software development, in all their forms”*

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


SCHOLARLY ECOSYSTEM

Aggregators

Publishers

Scholarly repositories

INDUSTRY

PUBLIC ADMINISTRATION

CULTURAL HERITAGE

Universal Software Archive

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

- connect scholarly ecosystem via Software Heritage
- use open non profit infrastructures
- open source first: *"all research software should be made available under an Open Source license by default"*

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

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

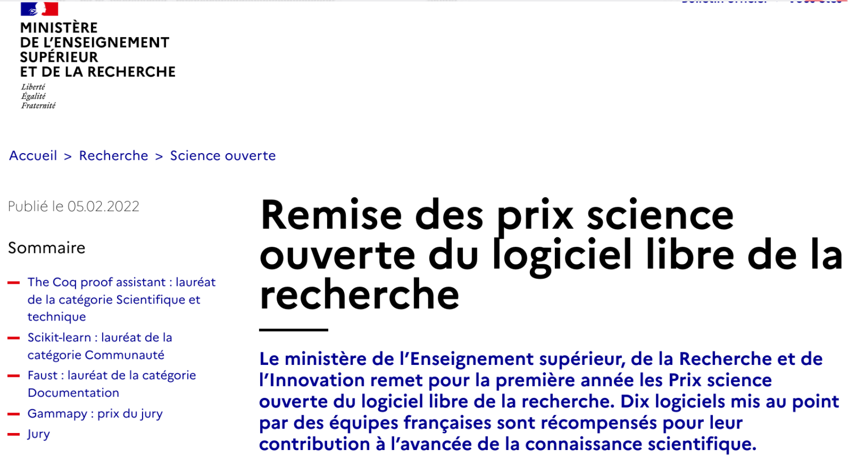
Open and promote research software source code

- actions (selection)
  - charter for research software policy
  - recognize software development (see announcement of the 2021 prize)
  - coordinate communities of practice
  - connected ecosystem of research outputs
- recommendations (selection)
  - archive in Software Heritage
  - standardise and use SWHID
  - build a national catalog of research software
  - leverage ADAC network

See official announcement

Meet the "Collège Logiciel" of the National Committee on Open Science (CoSO)!

Research Software is getting recognized



MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR
ET DE LA RECHERCHE

Liberté
Égalité
Fraternité

Accueil > Recherche > Science ouverte

Publié le 05.02.2022

Sommaire

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

Remise des prix science ouverte du logiciel libre de la recherche

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

Outline

1. Introduction
2. Open Science
3. An emerging policy framework
4. **Building the software pillar of Open Science: assessing the needs**
5. Focus on ARDC and infrastructures
6. Demo time!
7. Actions

A plurality of needs

Researchers

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

Laboratories/teams

- track software contributions
- produce reports
- maintain web page

Research Organization

know its software assets

- technology transfer
- impact metrics
- funding strategy
- career evaluation

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, . . .)
- Opening up towards a community (documentation, organization, communication)

Need training, best practices

Beyond ARDC

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

Outline

1. Introduction
2. Open Science
3. An emerging policy framework
4. Building the software pillar of Open Science: assessing the needs
5. **Focus on ARDC and infrastructures**
6. Demo time!
7. Actions

Where is the source code?

Collaborative development platforms (aka "forges")

- BitBucket, GitLab(.com), GitHub, etc.
- support for version control, issues, etc.
- example:
  - https://github.com/rdicosmo/parmap
  - https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/

Distribution platforms

- CTAN, CRAN, PyPi, Debian, etc.
- example: https://ctan.org/pkg/biblatex-software

Archives

- Software Heritage
- example: archived version of biblatex-software

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)

2019: big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases *250.000* repositories (including research software)

2021: … in Academia too

- october 2021: Inria's old gforge is unplugged
  - breaks the build chain of the OCaml package manager (Opam)

Bottomline

we need a universal archive of software source code: now we have one!

Software Heritage in a nutshell

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

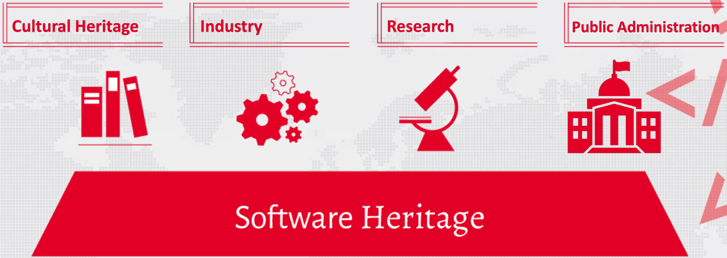
Universal archive

preserve all software source code

Research infrastructure

enable analysis of all software source code

The largest software archive, a shared infrastructure



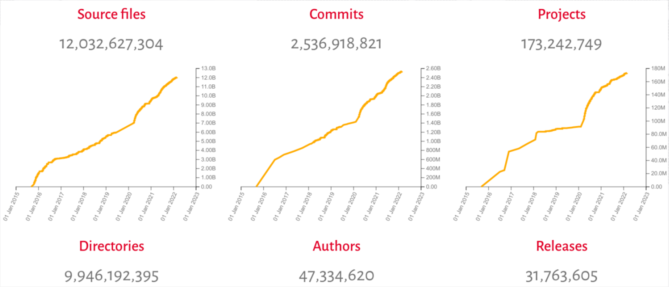
Cultural Heritage

Industry

Research

Public Administration

Software Heritage



Source files

12,032,627,304

Commits

2,536,918,821

Projects

173,242,749

Directories

9,946,192,395

Authors

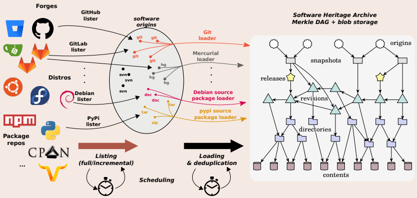
47,334,620

Releases

31,763,605

Addressing the four needs (see ICMS 2020 for details)

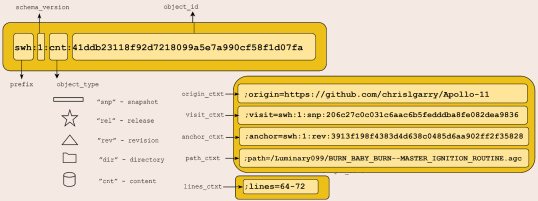
Archive (12B+ files, 170M+ projects)



- save.softwareheritage.org
- deposit.softwareheritage.org

Reference (20 billion SWHIDs)

Intrinsic, decentralised, cryptographically strong identifiers, SWHIDs


schema_version
object_id
swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa
prefix
object_type
"snp" - snapshot
"rel" - release
"rev" - revision
"dir" - directory
"cnt" - content
origin_ctxt
visit_ctxt
anchor_ctxt
path_ctxt
lines_ctxt
;origin=https://github.com/chrislgarry/Apollo-11
;visit=swh:1:snp:206c27c0c031c6aac6b5fedddba8fe082dea9836
;anchor=swh:1:rev:3913f198f4383d4d638c0485d6aa902ff2f35828
;path=/Luminary099/BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.agc
;lines=64-72

Now supported in SPDX 2.2, Wikidata etc.

Describe

- *Intrinsic metadata* from source code
- Contributed the Codemeta generator

Cite/Credit

- Contributed *software citation* style biblatex-software, v 1.2-2 now on CTAN

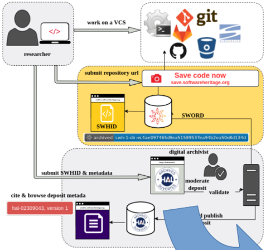
Outline

1. Introduction
2. Open Science
3. An emerging policy framework
4. Building the software pillar of Open Science: assessing the needs
5. Focus on ARDC and infrastructures
6. Demo time!
7. Actions

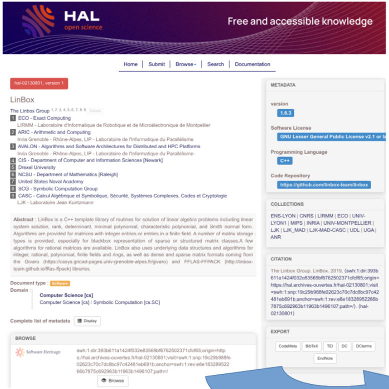
Demo time: a walkthrough

- Browse the archive (your work may be already there !)
- Trigger archival of your preferred software in a breeze
- Get and use SWHIDs (full specification available online)
- Cite software using the biblatex-software package from CTAN
- Example in a journal: an 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 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 a research article: compare Fig. 1 and conclusions
  - in the 2012 version
  - in the updated version using SWHIDs and Software Heritage
- Example use in a research article: extensive use of SWHIDs in a replication experiment

Overview of the Software Heritage / HAL synergy



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



HAL
open science
Free and accessible knowledge
Home | Submit | Browse | Search | Documentation
hal-02130801, version 1

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 Parallélisme
3 AVALON - Algorithms and Software Architectures for Distributed and HPC Platforms
Inria Grenoble - Rhône-Alpes, LIP - Laboratoire de l'Informatique du Parallélisme
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 Givaro (https://casys.gricad-pages.univ-grenoble-alpes.fr/givaro) 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]

Complete list of metadata Display

BROWSE

Software Heritage swh:1:dir:393b611a1424f032e83569bf6762502371cfcf65;origin=https://hal.archives-ouvertes.fr/hal-02130801;visit=swh:1:snp:19c29b988fe02623c70c7dc8bc97c42481eb691b;anchor=swh:1:rev:e8e18328952266b7875c692963b11963b1496107;path=/

Browse

METADATA

version
1.6.3

Software License
GNU Lesser General Public License v2.1 or la

Programming Language
C++

Code Repository
https://github.com/linbox-team/linbox

COLLECTIONS

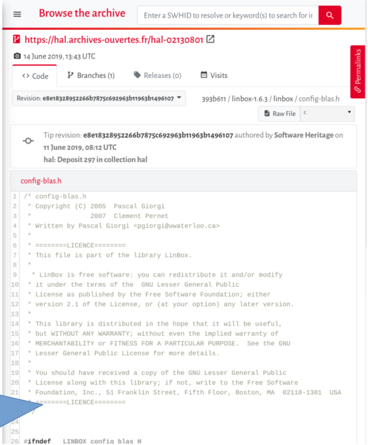
ENS-LYON | CNRS | LIRMM | ECO | UNIV-LYON1 | MIPS | INRIA | UNIV-MONTPELLIER | LJK | LJK_MAD | LJK-MAD-CASC | UDL | UGA | ANR

CITATION

The Linbox Group. LinBox. 2019, (swh:1:dir:393b611a1424f032e83569bf6762502371cfcf65;origin=https://hal.archives-ouvertes.fr/hal-02130801;visit=swh:1:snp:19c29b988fe02623c70c7dc8bc97c42481eb691b;anchor=swh:1:rev:e8e18328952266b7875c692963b11963b1496107;path=/). (hal-02130801)

EXPORT

CodeMeta | BibTeX | TEI | DC | DCterms | EndNote



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

14 June 2019, 13:43 UTC

Revision: e8e18328952266b7875c692963b11963b1496107

393b611 / linbox-1.6.3 / linbox / config-blas.h

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>
 *
 * ========LICENCE========
 * 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
 ========LICENCE========
 /


#ifndef __LINBOX_config_blas_H
```

swh:1:dir:393b611a1424f032e83569bf6762502371cfcf65

Outline

1. Introduction
2. Open Science
3. An emerging policy framework
4. Building the software pillar of Open Science: assessing the needs
5. Focus on ARDC and infrastructures
6. Demo time!
7. **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

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)