Software Heritage: a new infrastructure for Open Source and Open Science

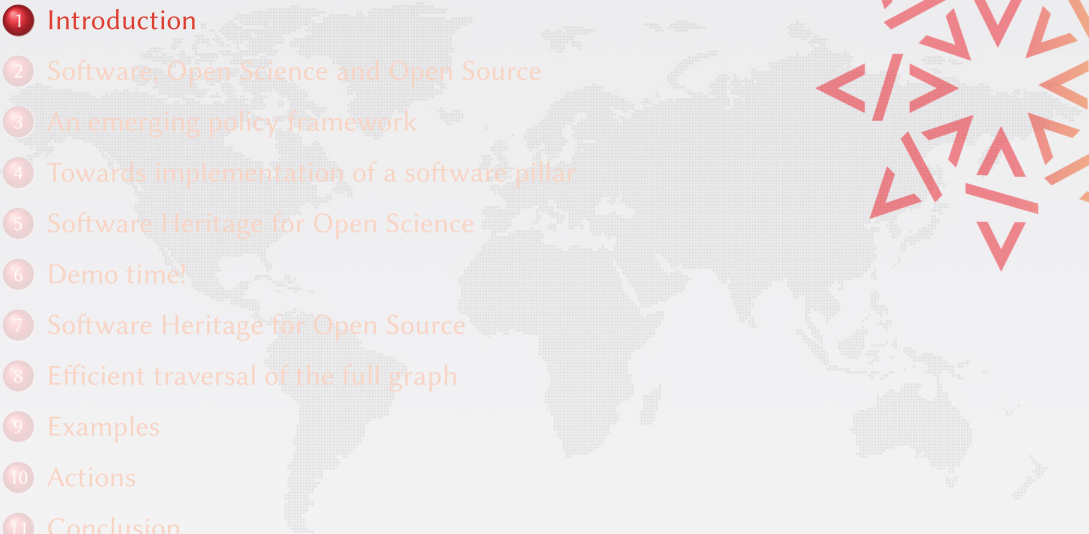
Roberto Di Cosmo
Conference at EPITA

Director, Software Heritage
Inria and Université de Paris Cité

October 21st 2022

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline



1. Introduction
2. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. Demo time!
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. Examples
10. Actions
11. Conclusion

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. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. Demo time!
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. Examples
10. Actions
11. Conclusion

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.”

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

19XX's compulsory exclusive copyright transfer to publishers (unlawful?) (notable exceptions: US federal agencies and UK Crown Copyright)

1990's Internet, Web and ArXiv break the marriage of convenience of researchers with publishers

2000's declarations (Budapest, 2001; Berlin 7, 2009) and actions (LIPIcs, 2009)

2010's reactions (SciHub, 2011; Plan S, 2018) and transformations (not so easy)

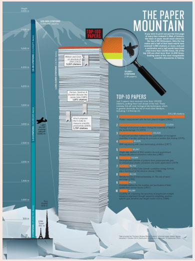
TL;DR: see my viewpoint in 2005 and the SIGPLAN blog in 2020

Open Data (less painful, but still unfinished story)

- 1957-1958: International Geophysical Year shows the way
- 2006 (and 2021): OECD recommendation on publicly funded research data
- 2016 and later: FAIR terminology (*focus on metadata, sort of forgets open…*)

Software: the third pillar of Open Science

Software powers modern research



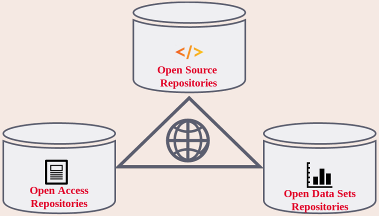
[. . .] software [. . .] essential in their fields.

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!

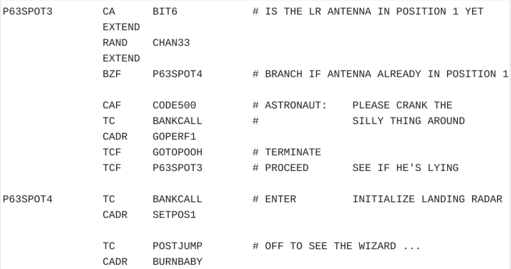
Preserving (the history of) source code is necessary for *reproducibility*

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)



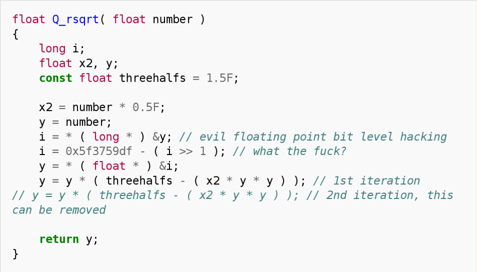
```
P63SPOT3        CA      BIT6            # IS THE LR ANTENNA IN POSITION 1 YET
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     P63SPOT4        # BRANCH IF ANTENNA ALREADY IN POSITION 1

                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."

Free and Open Source Software

Software that offers to *its users* the freedom to:

0. use the software
1. study and adapt the software
2. distribute software copies
3. distribute modified copies

Free Software has changed the way software is:

- developed
- tested
- deployed
- maintained
- marketed
- sold
- designed
- taught
- ...

Open Source vs. Free Software

Phylosophy

free software Richard Stallman
focus on user freedom

open source Bruce Perenes/Eric Raymond
focus on software development and reuse
Open Source Definition in 10 points

A long story

- formalised since the late ’80s
- existed long before

Licence spectrum

copylefted	GPL/LGPL, etc.
non copylefted	BSD/MIT, etc.

Outline

1. Introduction
2. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. Demo time!
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. Examples
10. Actions
11. Conclusion

International highlights

Paris Call on Software Source code (2019, UNESCO)

PARIS CALL
SOFTWARE SOURCE CODE
AS HERITAGE FOR SUSTAINABLE DEVELOPMENT

Software Heritage

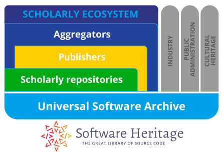
40 international experts call to *“promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, [. . .] recognising in the careers of academics their contributions to high quality software development, in all their forms”*

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

1

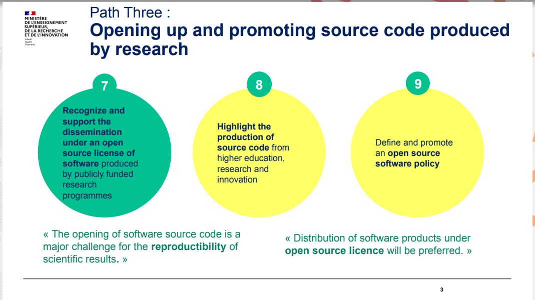
MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR,
DE LA RECHERCHE
ET DE L'INNOVATION

Second French Plan
for Open Science
GENERALISING
OPEN SCIENCE
IN FRANCE 2021-2024

Launch on 6 July 2021 by Frédérique Vidal, Minister for Higher Education, Research and Innovation

- Multiplying the **levers for change** in order to **generalise open science practices**
- Structuring the **policy for opening up or sharing research data**
- New commitments to the **opening of source code** produced by research
- **European and international inclusion** in the context of the French Presidency of the European Union
- **Disciplinary and thematic variations**: open science policies must be adapted to disciplinary specificities

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

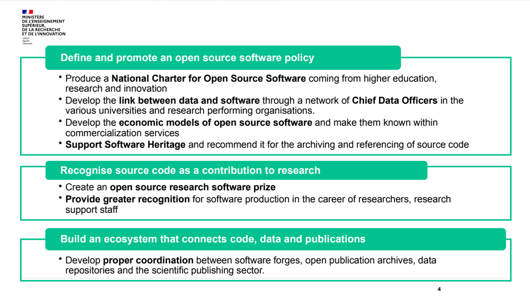
8 — **Highlight the production of source code** from higher education, research and innovation

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

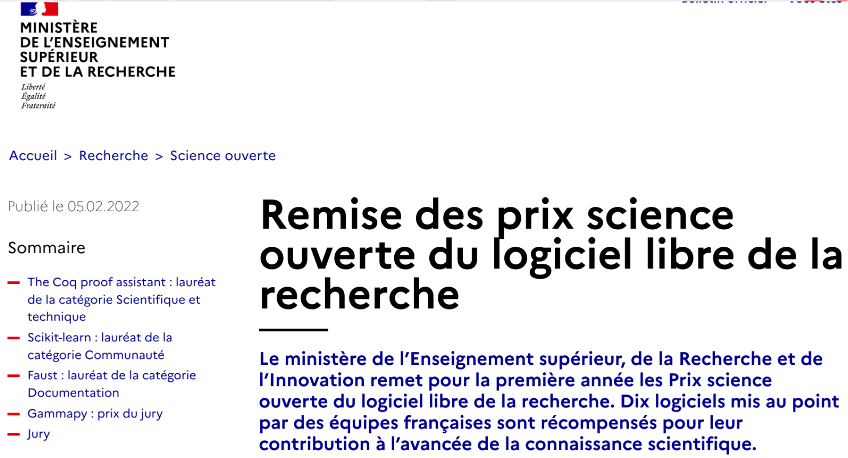
- Create an **open source research software prize**
- **Provide greater recognition** for software production in the career of researchers, research support staff

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

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. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. Demo time!
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. Examples
10. Actions
11. Conclusion

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.)

let's focus on infrastructures for ARDC

Approaches to preservation

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

.zip or .tar file on:

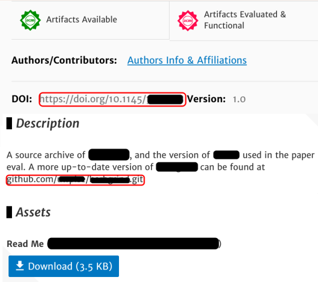
- ~~ftp server~~
- web page
- document archive (+ DOI)

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



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

Assets

Read Me [redacted]

⬇ Download (3.5 KB)

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

1. Introduction
2. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. **Software Heritage for Open Science**
6. Demo time!
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. Examples
10. Actions
11. Conclusion

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

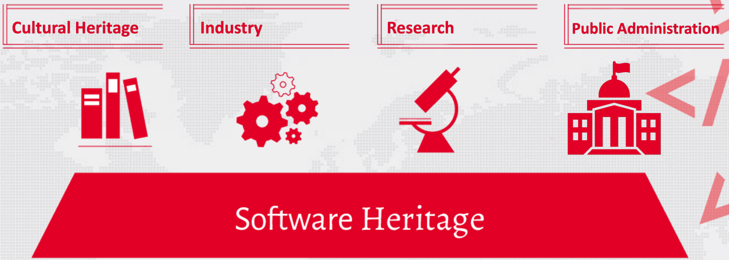
Universal archive

preserve all software source code

Research infrastructure

enable analysis of all software source code

The largest software archive, a shared infrastructure



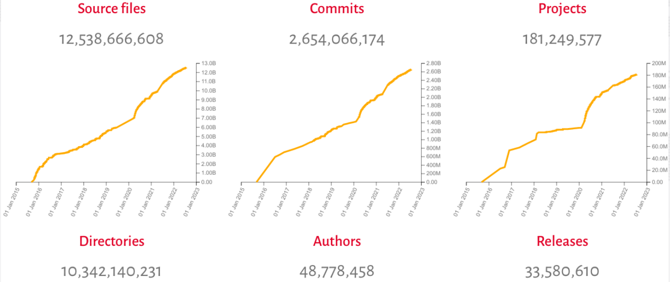
Cultural Heritage

Industry

Research

Public Administration

Software Heritage



Source files

12,538,666,608

Commits

2,654,066,174

Projects

181,249,577

Directories

10,342,140,231

Authors

48,778,458

Releases

33,580,610

An international, non profit initiative built for the long term

Sharing the vision

UNESCO
United Nations
Educational, Scientific and
Cultural Organization

IEEE Software · GitLab · ENGINEERING · eclipse

OW2 · fsfe · nlnet FOUNDATION · SIGSOFT · Liberté • Égalité • Fraternité RÉPUBLIQUE FRANÇAISE · INAPI Gobierno de Chile

THE LINUX FOUNDATION

ADULLACT · INFORMATICS EUROPE · Computer History Museum · software freedom conservancy

Software Freedom Law Center · AdaCore The GNAT Pro Company · gandi.net · TDF

creative commons · SIF Société Informatique de France · open source initiative · FREE SOFTWARE FOUNDATION · openinventionnetwork

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

Donors, members, sponsors

Inria
INVENTEURS DU MONDE NUMÉRIQUE

Diamond sponsor
cea

Platinum sponsors
cnrs · HUAWEI · intel
Microsoft · RÉPUBLIQUE FRANÇAISE MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR, DE LA RECHERCHE ET DE L'INNOVATION · SOCIETE GENERALE

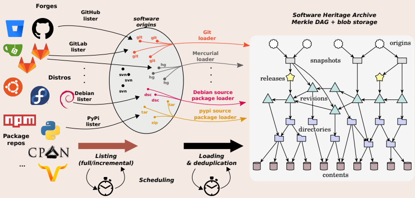
Gold sponsors
openinventionnetwork · Université de Paris · SORBONNE UNIVERSITÉ

Silver sponsors
AdaCore · Liberté • Égalité • Fraternité RÉPUBLIQUE FRANÇAISE · Google
GitHub · UNIVERSITÀ DI PISA · vmware

Bronze sponsors
DANS Data Archiving and Networked Services · FOSSID · Red Hat · ALMA MATER STUDIORUM A.D. 1088

Addressing the four ARDC needs (see ICMS 2020 for details)

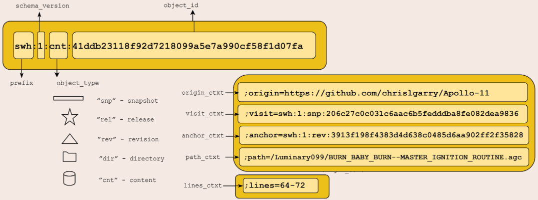
Archive (12B+ files, 180M+ projects)



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

Reference (25 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 ;origin=https://github.com/chrislgarry/Apollo-11

visit_ctxt ;visit=swh:1:snp:206c27c0c031c6aac6b5fedddba8fe082dea9836

anchor_ctxt ;anchor=swh:1:rev:3913f198f4383d4d638c0485d6aa902ff2f35828

path_ctxt ;path=/Luminary099/BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.agc

lines_ctxt ;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. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. **Demo time!**
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. Examples
10. Actions
11. Conclusion

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

Growing adoption of SWH in Academia (selection)

HAL software curated deposit workflow

Curated Archiving of Research Software Artifacts
International Journal of Digical Curation, 2020

Reference archive for swmath.org

swMATH
an information service for mathematical software

See *code* links, e.g.
SemiPar package

IPOL (image processing)

- archive (deposit)
- reference
- BibLaTeX

eLife (life sciences)

- archive (save code now)
- reference

JTCAM (mechanics)

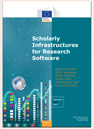
- instructions for authors
- biblatex-software in journal LaTeX class

Policy: France

NATIONAL PLAN FOR OPEN SCIENCE

National Plan for Open Science and Research Infrastructures

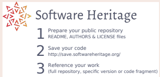
Policy: Europe


Scholarly Infrastructures for Research Software
Report from the EOSC Executive Board Working Group (WG) Architecture Task Force (TF) SIRS
Independent Expert Report

EOSC SIRS report

- SWHIDs
- archive

Guidelines


Software Heritage

1 Prepare your public repository
README, AUTHORS & LICENSE files

2 Save your code
http://save.softwareheritage.org/

3 Reference your work
(full repository, specific version or code fragment)

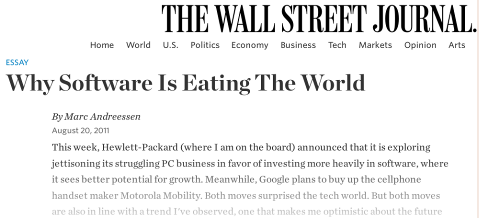
- summary
- ICMS 2020

Outline

1. Introduction
2. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. Demo time!
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. Examples
10. Actions
11. Conclusion

Open Source is growing…

Software is eating the world



THE WALL STREET JOURNAL.

Home World U.S. Politics Economy Business Tech Markets Opinion Arts

ESSAY

Why Software Is Eating The World

By Marc Andreessen

August 20, 2011

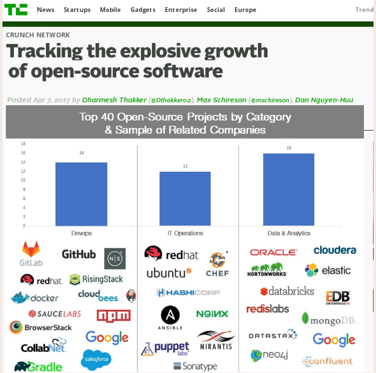
This week, Hewlett-Packard (where I am on the board) announced that it is exploring jettisoning its struggling PC business in favor of investing more heavily in software, where it sees better potential for growth. Meanwhile, Google plans to buy up the cellphone handset maker Motorola Mobility. Both moves surprised the tech world. But both moves are also in line with a trend I've observed, one that makes me optimistic about the future

Software companies outperform

or buy out traditional companies

Marc Andreesen, 2011

Open Source is eating the Software World



News Startups Mobile Gadgets Enterprise Social Europe Trend

CRUNCH NETWORK

Tracking the explosive growth of open-source software

Posted Apr 7, 2017 by Dharmesh Thakker (@Dthakkero2), Max Schireson (@mschireson), Dan Nguyen-Huu

Reuse is the new rule

80% to 90% of a new application is … just reuse! (Sonatype survey, 2017)

Improving Security and Transparency for Open Source

Where does reused software come from?

Sourceforge
Debian
CPAN
Gitorious
Maven
Inria
Bitbucket
GitHub
BerliOs
CTAN
GitLab
CRAN
GoogleCode
Adullact

Do *you* know where it comes from?

- the software you ship
- the software you use
- the software you acquire
- the software that
  - has that bug
  - has that vulnerability

KYSW: Know Your SoftWare

Like KYC in banking, KYSW is now essential all over IT…

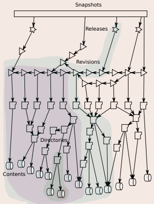
Sec. 4. Enhancing Software Supply Chain Security

ensuring and attesting, to the extent practicable, to the integrity and provenance of open source software

May 2021 POTUS Executive Order

A revolutionary infrastructure for industry

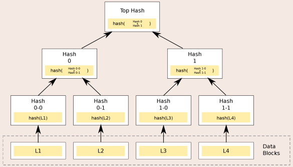
The *graph* of Software Development



All of the software development in a single graph!

- lookup by content hash
- wayback machine for software development
  - `http://archive.softwareheritage.org/`
- ... and much more

The *blockchain* of Software Development



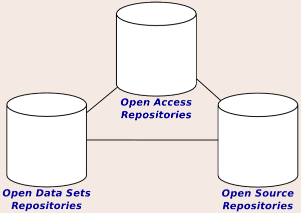
All of a software development... in a single Merkle graph!

Widely used crypto (e.g., Git, blockchains, IPFS, ...)

- built-in deduplication
- intrinsic, unforgeable identifiers at all levels
- simplifies traceability (licensing, supply chain management)

A revolutionary infrastructure for science

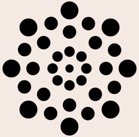
A *pillar* of Open Science



The *reference archive* of Research Software for Open Science

- curated deposit of research software
  - in collaboration with HAL, CCSD and Inria IES
  - now open *to all researchers*!
- intrinsic identifiers for reproducibility

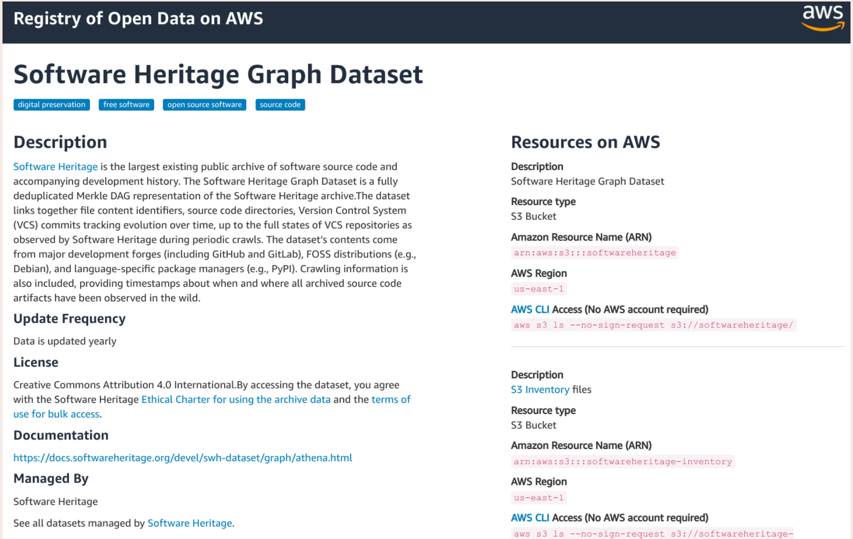
Reference platform for *Big Code*



- unique observatory of all software development
- big data, machine learning paradise: classification, trends, coding patterns, code completion…

The full graph in the AWS Open Data collection

https://registry.opendata.aws/software-heritage/



Registry of Open Data on AWS

Software Heritage Graph Dataset

digital preservation | free software | open source software | source code

Description

Software Heritage is the largest existing public archive of software source code and accompanying development history. The Software Heritage Graph Dataset is a fully deduplicated Merkle DAG representation of the Software Heritage archive.The dataset links together file content identifiers, source code directories, Version Control System (VCS) commits tracking evolution over time, up to the full states of VCS repositories as observed by Software Heritage during periodic crawls. The dataset's contents come from major development forges (including GitHub and GitLab), FOSS distributions (e.g., Debian), and language-specific package managers (e.g., PyPI). Crawling information is also included, providing timestamps about when and where all archived source code artifacts have been observed in the wild.

Update Frequency

Data is updated yearly

License

Creative Commons Attribution 4.0 International.By accessing the dataset, you agree with the Software Heritage Ethical Charter for using the archive data and the terms of use for bulk access.

Documentation

https://docs.softwareheritage.org/devel/swh-dataset/graph/athena.html

Managed By

Software Heritage

See all datasets managed by Software Heritage.

Resources on AWS

Description
Software Heritage Graph Dataset

Resource type
S3 Bucket

Amazon Resource Name (ARN)
`arn:aws:s3:::softwareheritage`

AWS Region
`us-east-1`

AWS CLI Access (No AWS account required)

```
aws s3 ls --no-sign-request s3://softwareheritage/
```

Description
S3 Inventory files

Resource type
S3 Bucket

Amazon Resource Name (ARN)
`arn:aws:s3:::softwareheritage-inventory`

AWS Region
`us-east-1`

AWS CLI Access (No AWS account required)

```
aws s3 ls --no-sign-request s3://softwareheritage-
```

A peek at the dataset

Accessing graph leaves (a.k.a. contents)

```
$ aws s3 ls --no-sign-request s3://softwareheritage/
         PRE content/
         PRE graph/
```

File contents can be accessed using their SHA1 checksum

```
$ aws s3 cp --no-sign-request \
  s3://softwareheritage/content/8624bcdae55baeef00cd11d5dfcfa60f68710a02 .
```

Notice that file contents are compressed:

```
$ zcat 8624bcdae55baeef00cd11d5dfcfa60f68710a02 | head
  GNU GENERAL PUBLIC LICENSE
     Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

A peek at the dataset, cont'd

Annual dumps of (inner nodes of) the full graph

```
$ aws s3 ls --no-sign-request s3://softwareheritage/graph/
    PRE 2018-09-25/
    PRE 2019-01-28-popular-3k-python/                 PRE 2021-03-23-cpython-3-5/
    PRE 2019-01-28-popular-4k/                        PRE 2021-03-23-popular-3k-python/
    PRE 2020-05-20/                                   PRE 2021-03-23/
    PRE 2020-12-15/                                   PRE 2022-04-25/
```

How to use (there is much more, e.g. swh-graph!) and cite

- online full documentation, and read Antoine Pietri's PhD Thesis
- Antoine Pietri, Diomidis Spinellis, Stefano Zacchiroli. *The Software Heritage Graph Dataset: Public software development under one roof.* MSR 2019. (bibtex)

A game changer for ESE studies

- broad variety of sources (reduce GH bias) in *one open dataset*
- one reference *standard data format* (VCS are abstracted away)
- greatly simplifies reproducibility packages (*just list the SWHIDs!*)

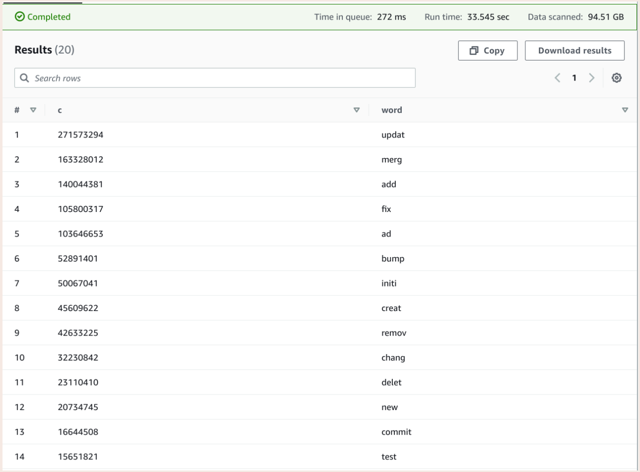
Example: most popular commit verbs (stemmed)

Query using Amazon Athena

```
SELECT COUNT(*) AS C, word FROM (
   SELECT word_stem(lower(split_part(
    trim(from_utf8(message)),’ ’, 1)))
   AS word FROM revision
   WHERE length(message) < 1000000)
WHERE word != ’’
GROUP BY word
ORDER BY C
DESC LIMIT 20;
```

Total cost: approximately .5 euros

Results



✓ Completed   Time in queue: 272 ms   Run time: 33.545 sec   Data scanned: 94.51 GB

Results (20)

#	c	word
1	271573294	updat
2	163328012	merg
3	140044381	add
4	105800317	fix
5	103646653	ad
6	52891401	bump
7	50067041	initi
8	45609622	creat
9	42633225	remov
10	32230842	chang
11	23110410	delet
12	20734745	new
13	16644508	commit
14	15651821	test

Outline

1. Introduction
2. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. Demo time!
7. Software Heritage for Open Source
8. **Efficient traversal of the full graph**
9. Examples
10. Actions
11. Conclusion

Going beyond SQL

State-of-the-art graph compression from social networks

Paolo Boldi, Antoine Pietri, Sebastiano Vigna, Stefano Zacchiroli
Ultra-Large-Scale Repository Analysis via Graph Compression
SANER 2020, 27th Intl. Conf. on Software Analysis, Evolution and Reengineering. IEEE

Results

Full graph structure (25 B nodes, 350 B edges) in 200 GiB RAM

- traversal time is tens of ns per edge
- bidirectional traversals implemented
- beware: metadata access is still *off RAM*

Java and gRPC APIs available

docs.softwareheritage.org/devel/swh-graph/grpc-api.html

Examples

assume graph service on localhost:50091

Find all origins containing a given content

```
grpc_cli call localhost:50091 swh.graph.TraversalService.Traverse "\
src: ’swh:1:cnt:8722d84d658e5e11519b807abb5c05bfbfc531f0’, direction: BACKWARD, \
mask: {paths: [’swhid’,’ori.url’]}, return_nodes: {types: ’ori’}"
```

Gives a list of origins including "`https://github.com/rdicosmo/parmap`", encoded as "swh:1:ori:8903a90cff8f07159be7aed69f19d66d33db3f86" (beware: this is not a SWHID!)

Shortest provenance path of a content in a given origin

```
grpc_cli call localhost:50091 swh.graph.TraversalService.FindPathBetween "\
src: ’swh:1:ori:8903a90cff8f07159be7aed69f19d66d33db3f86’, \
dst: ’swh:1:cnt:8722d84d658e5e11519b807abb5c05bfbfc531f0’, \
mask: {paths: [’swhid’]}" | egrep ’swhid’
connecting to localhost:50091
  swhid: "swh:1:ori:8903a90cff8f07159be7aed69f19d66d33db3f86"
  swhid: "swh:1:snp:1527a93b039d70f6a781b05d76b77c6209912887"
  swhid: "swh:1:rev:82df563aecf86b9164eee7d10d40f2d8cbd1c78d"
  swhid: "swh:1:dir:484db39bb282588619l837bb0960b7450f9099bb"
  swhid: "swh:1:dir:4d15e44b378fe39dd23817abee756cd47ad14575"
  swhid: "swh:1:cnt:8722d84d658e5e11519b807abb5c05bfbfc531f0"
Rpc succeeded with OK status
```

Outline

1. Introduction
2. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. Demo time!
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. **Examples**
10. Actions
11. Conclusion

Selected research works using Software Heritage

- Thibault Allançon, Antoine Pietri, Stefano Zacchiroli
  The Software Heritage Filesystem (SwhFS): Integrating Source Code Archival with Development.
  ICSE 2021: The 43rd International Conference on Software Engineering https://arxiv.org/abs/2102.06390

- Stefano Zacchiroli
  Gender Differences in Public Code Contributions: a 50-year Perspective
  IEEE Softw. 38(2): 45-50 (2021)

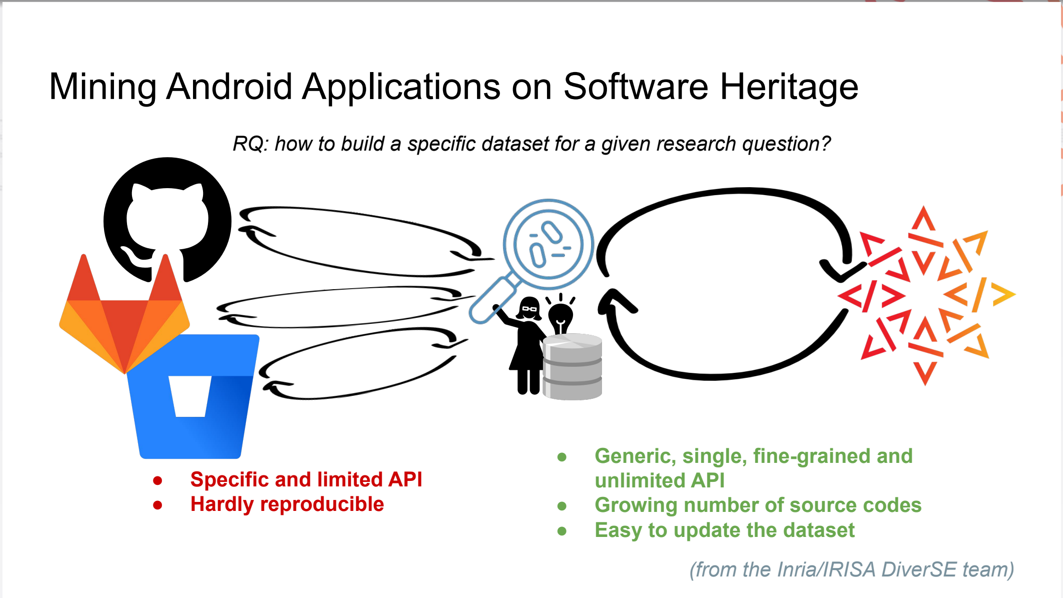
- Antoine Pietri, Guillaume Rousseau, Stefano Zacchiroli
  Forking Without Clicking: on How to Identify Software Repository Forks
  MSR 2020: 17th Intl. Conf. on Mining Software Repositories. IEEE

- Antoine Pietri, Guillaume Rousseau, Stefano Zacchiroli
  Determining the Intrinsic Structure of Public Software Development History
  MSR 2020: 17th Intl. Conf. on Mining Software Repositories. IEEE

- Paolo Boldi, Antoine Pietri, Sebastiano Vigna, Stefano Zacchiroli
  Ultra-Large-Scale Repository Analysis via Graph Compression
  SANER 2020, 27th Intl. Conf. on Software Analysis, Evolution and Reengineering. IEEE

- Roberto Di Cosmo, Guillaume Rousseau, Stefano Zacchiroli
  Software Provenance Tracking at the Scale of Public Source Code
  Empirical Software Engineering 25(4): 2930-2959 (2020)

From fellow researchers (Inria DiverSE team, thanks Benoît Combemale)



Mining Android Applications on Software Heritage

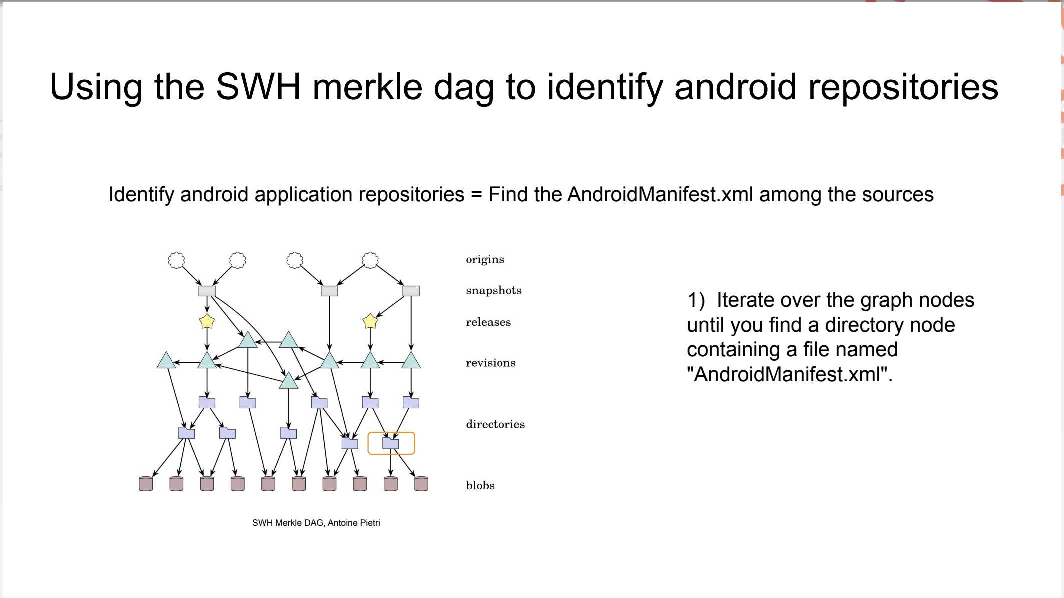
RQ: how to build a specific dataset for a given research question?

- Specific and limited API
- Hardly reproducible

- Generic, single, fine-grained and unlimited API
- Growing number of source codes
- Easy to update the dataset

(from the Inria/IRISA DiverSE team)

From fellow researchers (Inria DiverSE team, thanks Benoît Combemale)



Using the SWH merkle dag to identify android repositories

Identify android application repositories = Find the AndroidManifest.xml among the sources

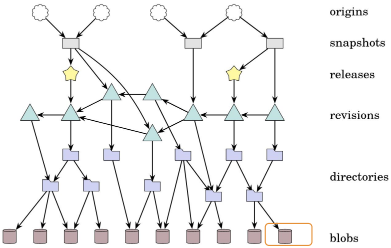
SWH Merkle DAG, Antoine Pietri

1) Iterate over the graph nodes until you find a directory node containing a file named "AndroidManifest.xml".

From fellow researchers (Inria DiverSE team, thanks Benoît Combemale)

Using the SWH merkle dag to identify android repositories

Identify android application repositories = Find the AndroidManifest.xml among the sources



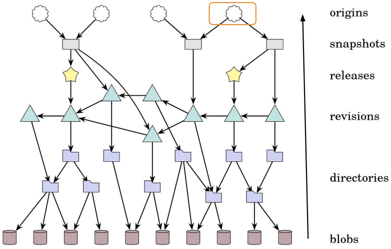
SWH Merkle DAG, Antoine Pietri

2) Extract the SWH identifier of the blob corresponding to the AndroidManifest.xml and download the corresponding file through the SWH Web API

From fellow researchers (Inria DiverSE team, thanks Benoît Combemale)

Using the SWH merkle dag to identify android repositories

Identify android application repositories = Find the AndroidManifest.xml among the sources



SWH Merkle DAG, Antoine Pietri

3) Traverse the graph in backward direction to the origin node and get the repository url

Bottomline

Broad variety of sources in *one open dataset*

reduces usual GH bias

Reference simple *standard data format*

VCS and forge details are abstracted away

Simplifies reproducibility packages

no need to create a full copy, *just list the SWHIDs!*

Software Heritage does the heavy lifting for you

no need to scrape/download repositories all over again

Outline

1. Introduction
2. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. Demo time!
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. Examples
10. **Actions**
11. Conclusion

A rally flag for a grand vision

Bring together academia, industry, governments, communities

"to build a reference, global infrastructure for open and better software"

Software Heritage is the first brick …

- vendor neutral
- open source
- a worldwide initiative
- a long term initiative

… that will enable

- archival, reference, integrity
- qualification, sharing and reuse
- a global software knowledge base
- test and deploy world class tooling

A lot more is needed

Software Heritage can be the *catalyser* of a way bigger undertaking

You can help

adopt *use* SWH in your work

- archive (research) software in SWH
- reference it using the SWHID *identifiers*

save relevant source code

contribute it's open source!

advocate spread the word, become and ambassador

research tackle scientific challenges

building SWH graph queries, efficient storage, distributed archival, classification, search, ...

using SWH the *Software Heritage graph dataset*

Outline

1. Introduction
2. Software, Open Science and Open Source
3. An emerging policy framework
4. Towards implementation of a software pillar
5. Software Heritage for Open Science
6. Demo time!
7. Software Heritage for Open Source
8. Efficient traversal of the full graph
9. Examples
10. Actions
11. **Conclusion**

Join the revolution!

www.softwareheritage.org @swheritage

Library of Alexandria of code

- recover the past
- structure the future

A CERN for Software

- build better software
  - for industry
  - for society as a whole

Jean-François Abramatic, Roberto Di Cosmo, Stefano Zacchiroli
Building the Universal Archive of Source Code
Communications of the ACM, October 2018

Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli
Identifiers for Digital Objects: the Case of Software Source Code Preservation
iPRES 2018: Intl. Conf. on Digital Preservation