

Securing the (Open Source) Software Supply Chain and the Software Heritage infrastructure

Roberto Di Cosmo

Director, Software Heritage
Inria and Université de Paris Cité

January 26th 2024



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

- 1 Introduction
- 2 The (open source) software tidal wave
- 3 (Open Source) Software Supply Chain
- 4 Meet Software Heritage
- 5 Meet the Software Heritage dataset(s)
- 6 Efficient traversal of the full graph
- 7 Demo time!
- 8 Impact on ESE studies
- 9 Call to action



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

Software *Source Code* is Precious Knowledge

Harold Abelson, Structure and Interpretation of Computer Programs (1st ed.)

1985

“Programs must be written for people to read, and only incidentally for machines to execute.”

Apollo 11 source code (excerpt)

```
P63SP0T3      CA      BIT6      # IS THE LR ANTENNA IN POSITION 1 YET
               EXTEND
               RAND      CHAN33
               EXTEND
               BZF      P63SP0T4      # BRANCH IF ANTENNA ALREADY IN POSITION 1

               CAF      CODE500      # ASTRONAUT: PLEASE CRANK THE
               TC      BANKCALL      # SILLY THING AROUND
               CADR      GOPERF1
               TCF      GOTOP00H      # TERMINATE
               TCF      P63SP0T3      # PROCEED SEE IF HE'S LYING

P63SP0T4      TC      BANKCALL      # ENTER INITIALIZE LANDING RADAR
               CADR      SETPOS1

               TC      POSTJUMP      # OFF TO SEE THE WIZARD ...
               CADR      BURNBABY
```

Quake III source code (excerpt)

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this
    // can be removed

    return y;
}
```

Len Shustek, Computer History Museum

2006

“Source code provides a view into the mind of the designer.”

~ 50 years, a lightning fast growth

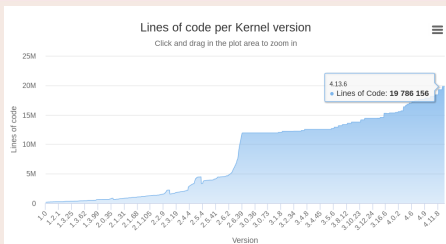
Apollo 11 Guidance Computer (~60.000 lines), 1969



"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



... now in your pockets!

Outline

- 
- 1 Introduction
 - 2 The (open source) software tidal wave
 - 3 (Open Source) Software Supply Chain
 - 4 Meet Software Heritage
 - 5 Meet the Software Heritage dataset(s)
 - 6 Efficient traversal of the full graph
 - 7 Demo time!
 - 8 Impact on ESE studies
 - 9 Call to action

Software is eating the world...

Business

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

hardware companies

Marc Andreessen, 2011

Technology

Software Defined Everything

Hardware gets commoditised

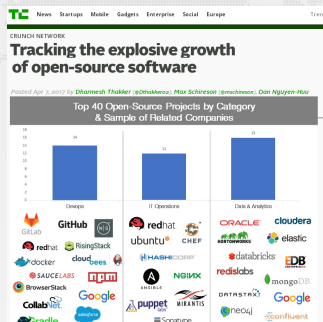
Software becomes the new value!



Open Source is eating the software world

Open Source Software

can be openly (re)used, modified, (re)distributed, *with full access to its source code!*



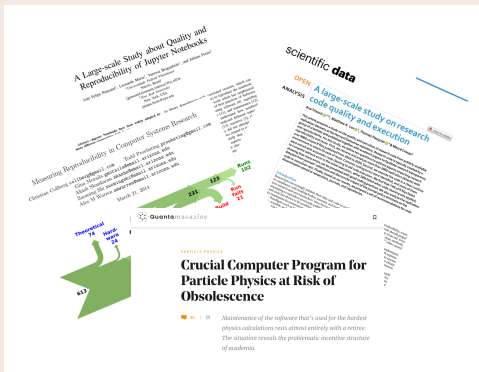
Reuse is the new rule

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

(Sonatype survey, 2017)

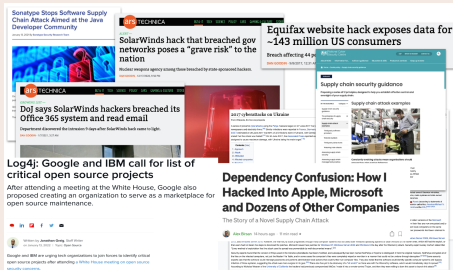
How are we managing our (open source) software ?

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

Policy highlights: Open Science

Paris Call on Software Source code (2019, UNESCO)

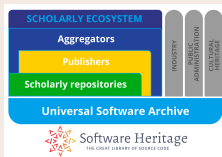


40 international experts call to “promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, [...] recognising in the careers of academics their contributions to high quality software development, in all their forms”



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

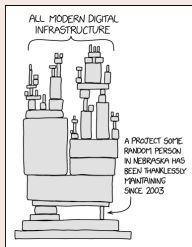
Software track in [OSEC 2022](#), Software working group launched in Science Europe, DFG adds software [to model CV \(9/22\)](#), NASA unveils [Open Science policy \(12/22\)](#), ...

Outline

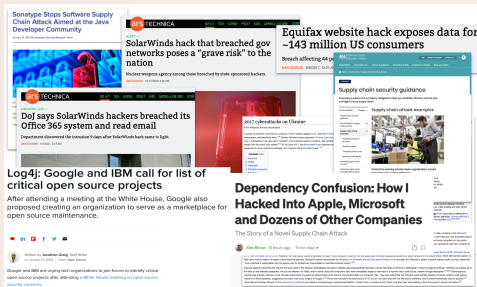
- 
- 1 Introduction
 - 2 The (open source) software tidal wave
 - 3 (Open Source) Software Supply Chain
 - 4 Meet Software Heritage
 - 5 Meet the Software Heritage dataset(s)
 - 6 Efficient traversal of the full graph
 - 7 Demo time!
 - 8 Impact on ESE studies
 - 9 Call to action

Software supply chain and its issues

Complex digital infrastructure



Software supply chain in the news



Software Supply Chain attacks

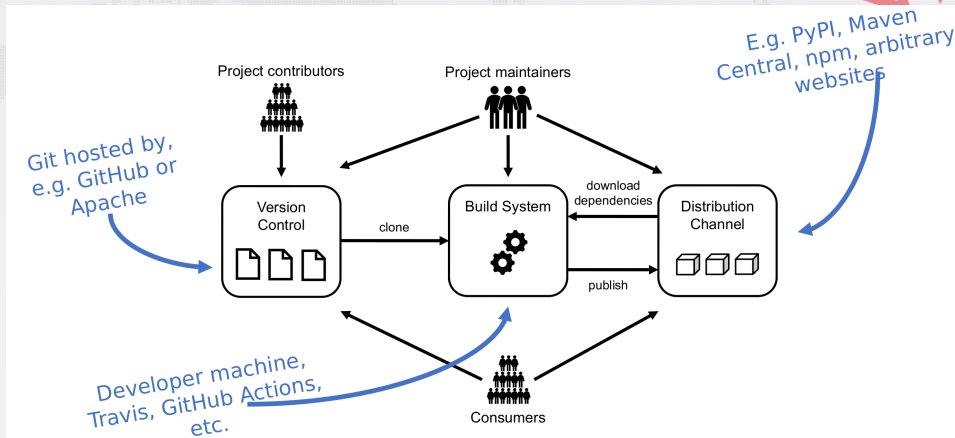
Malicious code injection into software components to compromise downstream users

March 2022 node-ipc and peacenotwar (CVE-2022-23812)

Dec 2021 Apache Log4j Remote Code Execution (Log4Shell, CVE-2021-44228)

Nov 2018 Attack on NPM package event-stream

Software supply chain in a picture



Policy highlights: industry and sovereignty

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

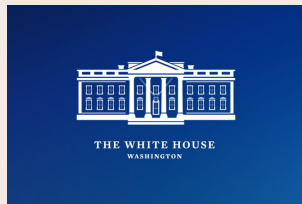
Vertical approach: Secure Your Software



improve security of *each component* separately

- by law: e.g. EU [Cyber Resilience Act](#)
- by practice: e.g. <https://best.openssf.org/>

Horizontal approach: all the supply chain



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 long road ahead

Vertical approach

secure *each component* separately

Horizontal approach

explore *the whole supply chain*

A few key challenging properties

findability needs **qualified metadata**

availability needs **an archive** and a **system of identifiers**

integrity needs **crypto**

traceability needs **a global provenance database**

reproducibility needs **groundbreaking tools**

these are relevant for Open Science too

We need a *global coordinated effort...*

and a common, open, shared infrastructure to track all (Open Source) software!

Forges are key platforms, but they are *not* enough!

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

source code is spread across hundreds of them...

lack of uniformity, no persistence guarantee

Outline

- 1 Introduction
- 2 The (open source) software tidal wave
- 3 (Open Source) Software Supply Chain
- 4 Meet Software Heritage**
- 5 Meet the Software Heritage dataset(s)
- 6 Efficient traversal of the full graph
- 7 Demo time!
- 8 Impact on ESE studies
- 9 Call to action





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

The largest software archive, a shared infrastructure

One infrastructure
open and shared

Cultural Heritage



Industry



Research



Public Administration

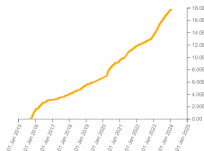


Software Heritage

The largest archive ever built

Source files

17,798,218,376

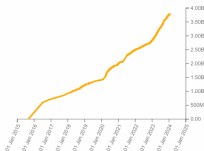


Directories

14,364,868,206

Commits

3,802,143,973

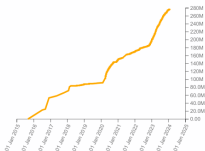


Authors

69,923,710

Projects

278,187,495



Releases

82,196,102

Bitbucket

2,509,402 origins



56,983 origins

git

24,600 origins



26,599 origins



136,338 origins



53,297 origins

GitHub

197,883,004 origins

gitle

10,171 origins

GitLab

4,216,298 origins



2,926 origins



172 origins



971,549 origins



14,482 origins



354 origins



1,207 origins

launchpad

503,631 origins

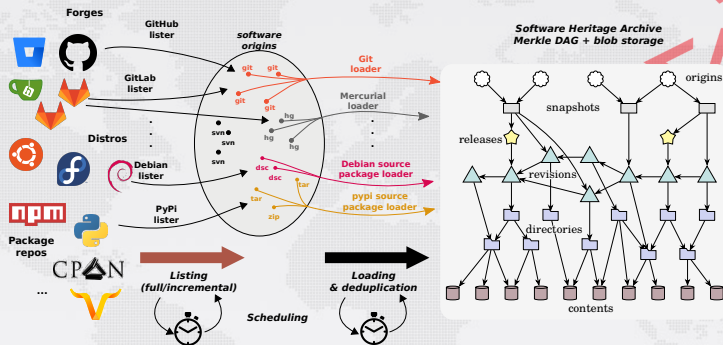
Maven

312,461 origins



14,482 origins

Address common Open Science and Open Source needs: archival




Global development history permanently archived in a uniform data model

- over 17 billion unique source files from over 270 million software projects
- ~1.5PB (compressed) blobs, ~35 B nodes, ~500 B edges

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

Supported listers ([index](#))



Software Heritage - User Documentation

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Search docs

CONTENTS:
Frequently Asked Questions

Software Heritage listers





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

» Software Heritage listers [View page source](#)

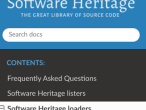
Software Heritage listers

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

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

Lister name	Related links	Current status	Related grants
 Arch lister	<ul style="list-style-type: none">Source codeDevelopment	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 AUR lister	<ul style="list-style-type: none">Source codeDevelopment	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 Bitbucket lister	<ul style="list-style-type: none">Source codeDeveloper docDevelopment	in production	
 Bower lister	<ul style="list-style-type: none">Source codeDevelopment	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	<ul style="list-style-type: none">Source codeDevelopment	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 Archive loader	<ul style="list-style-type: none">Source codeDeveloper doc	in production	
 AUR loader	<ul style="list-style-type: none">Source codeDevelopment	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 Bazaar loader	<ul style="list-style-type: none">Source codeDeveloper docDevelopment	in production	Alfred P. Sloan Foundation (awarded to Octobus)
	<ul style="list-style-type: none">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](#)



35+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](#)

Sharing the vision



And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Inria

Diamond sponsor



Platinum sponsors



Gold sponsors

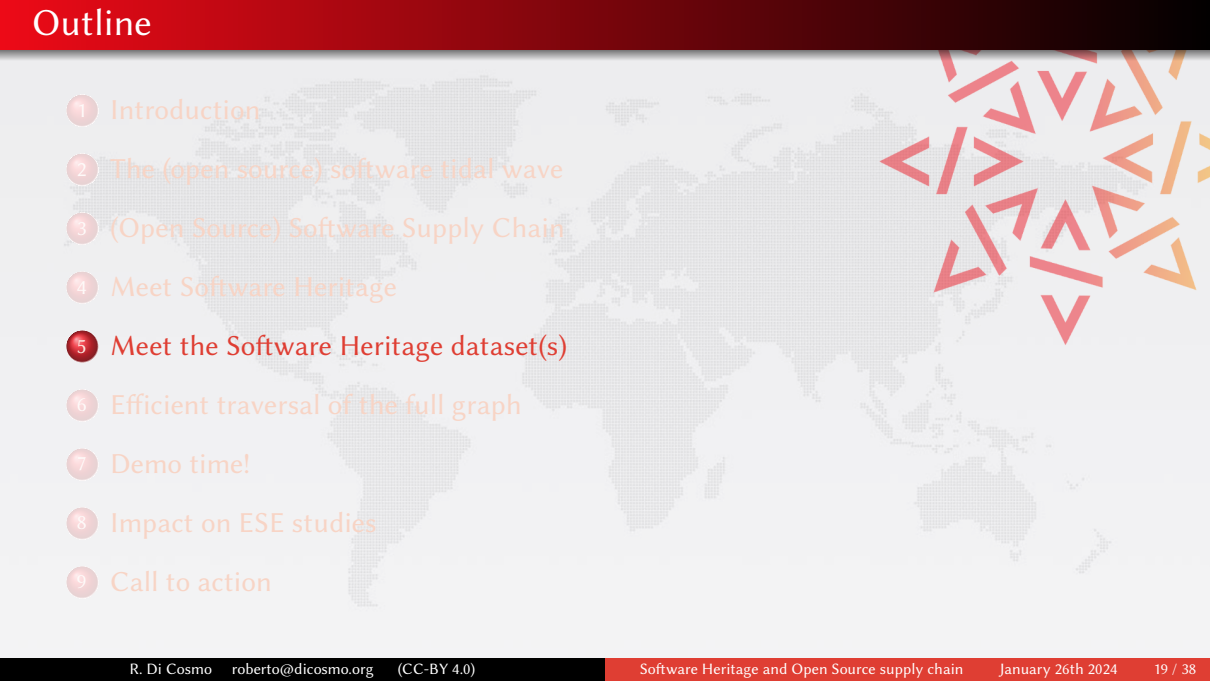


Silver sponsors



Bronze sponsors



- 
- 1 Introduction
 - 2 The (open source) software tidal wave
 - 3 (Open Source) Software Supply Chain
 - 4 Meet Software Heritage
 - 5 Meet the Software Heritage dataset(s)**
 - 6 Efficient traversal of the full graph
 - 7 Demo time!
 - 8 Impact on ESE studies
 - 9 Call to action

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-
```

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/
```

```
2018-09-25/
```

```
2019-01-28-popular-3k-python/
```

```
2019-01-28-popular-4k/
```

```
2020-05-20/
```

```
2020-12-15/
```

```
2021-03-23-cpython-3-5/
```

```
2021-03-23-popular-3k-python/
```

```
2021-03-23/
```

```
2022-04-25/
```

How to use

- [online full documentation](#)
- [Antoine Pietri's PhD Thesis](#)

How to cite

Antoine Pietri, Diomidis Spinellis, Stefano Zacchiroli. *The Software Heritage Graph Dataset: Public software development under one roof*. MSR 2019. ([bibtex](#))

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

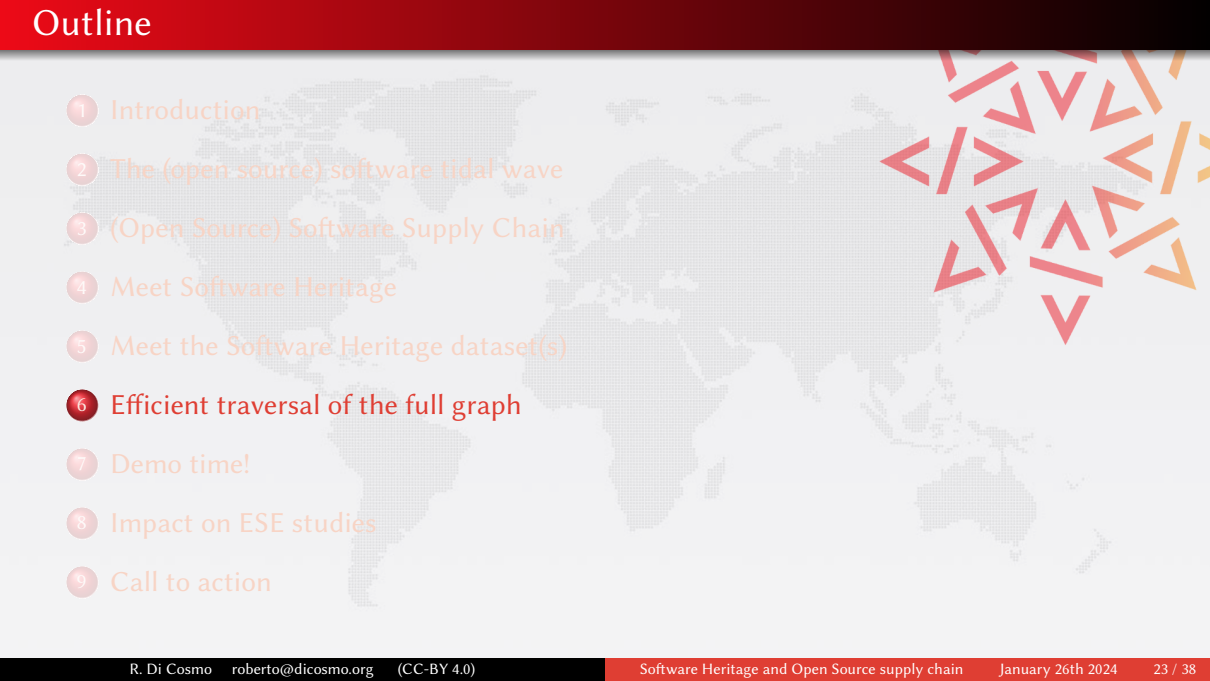
Results (20)

Copy

Download results

Search rows

#	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

- 
- 1 Introduction
 - 2 The (open source) software tidal wave
 - 3 (Open Source) Software Supply Chain
 - 4 Meet Software Heritage
 - 5 Meet the Software Heritage dataset(s)
 - 6 Efficient traversal of the full graph**
 - 7 Demo time!
 - 8 Impact on ESE studies
 - 9 Call to action

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 (35 B nodes, 500 B edges) in 300 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 ... Rust is coming next!

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

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"

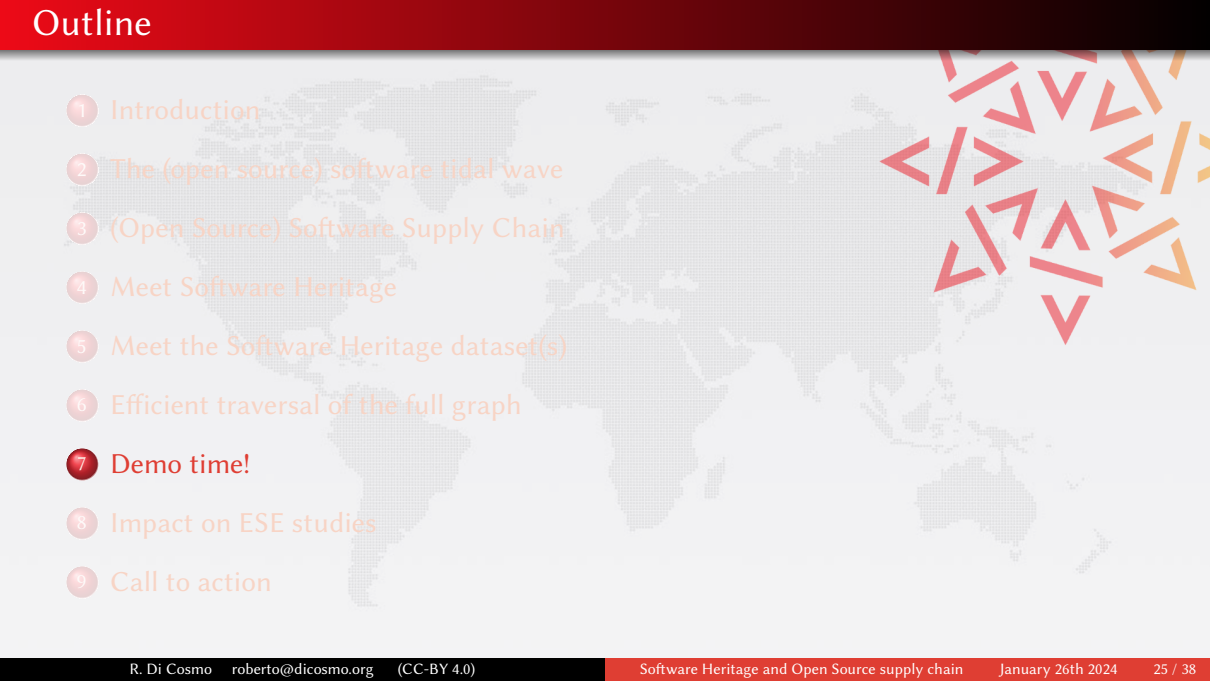
swhid: "swh:1:rev:82df563aecf86b9164eee7d10d40f2d8cbd1c78d"

swhid: "swh:1:dir:484db39bb2825886191837bb0960b7450f9099bb"

swhid: "swh:1:dir:4d15e44b378fe39dd23817abee756cd47ad14575"

swhid: "swh:1:cnt:8722d84d658e5e11519b807abb5c05bfbfc531f0"

Rpc succeeded with OK status

- 
- 1 Introduction
 - 2 The (open source) software tidal wave
 - 3 (Open Source) Software Supply Chain
 - 4 Meet Software Heritage
 - 5 Meet the Software Heritage dataset(s)
 - 6 Efficient traversal of the full graph
 - 7 Demo time!
 - 8 Impact on ESE studies
 - 9 Call to action

Get SWHID of compromised openssl files

```
$ for f in openssl-1.0.1*/ssl/d1_both.c; do swh-identify $f; done
swh:1:cnt:0a84f957118afa9804451add380eca4719a9765e  openssl-1.0.1-beta1/ssl/d1_both
swh:1:cnt:7a5596a6b373aeabbd6d8d674f0e20b1618c5012  openssl-1.0.1f/ssl/d1_both.c
swh:1:cnt:2e8cf681ed0976e2b16460170fda27c77cfec6cc  openssl-1.0.1g/ssl/d1_both.c
swh:1:cnt:04aa23107ec53c184505e98091306c7391091bb5  openssl-1.0.1h/ssl/d1_both.c
swh:1:cnt:de8bab873f2cf114d0d1b3e49acfa09bb9d0e4f7  openssl-1.0.1/ssl/d1_both.c
```

Look up one origin that contains it using the graph

```
$ swh-graph-lookup.py -c swh:1:cnt:de8bab873f2cf114d0d1b3e49acfa09bb9d0e4f7
swh:1:cnt:de8bab873f2cf114d0d1b3e49acfa09bb9d0e4f7;
path=ssl/d1_both.c;
anchor=swh:1:rev:86628df45f9eec5b2d46aeb77644ae8f544d1291;
visit=swh:1:snp:6163a539c30011303b5162931fdafd84af8d1c09;
origin=https://github.com/taptipalit/openssl
```

let's check [this occurrence](#)

... more closely

Find all origins...

```
$ swh-graph-lookup.py --all-origins \  
  -c swh:1:cnt:de8bab873f2cf114d0d1b3e49acfa09bb9d0e4f7 \  
  | cut -d ; -f 3 | sort -u | grep swh | sed 's/anchor=/' > allrevs  
$ head -3 allrevs  
swh:1:rev:005b61176f3f72c6c31a2c9431dbc8b5730023ed  
swh:1:rev:01b47383d76f8b9653c6418b0fe1c36043b83ea1  
swh:1:rev:03487116266297d1611556910515f7a3cd7f5fcd
```

One of them is pretty late!

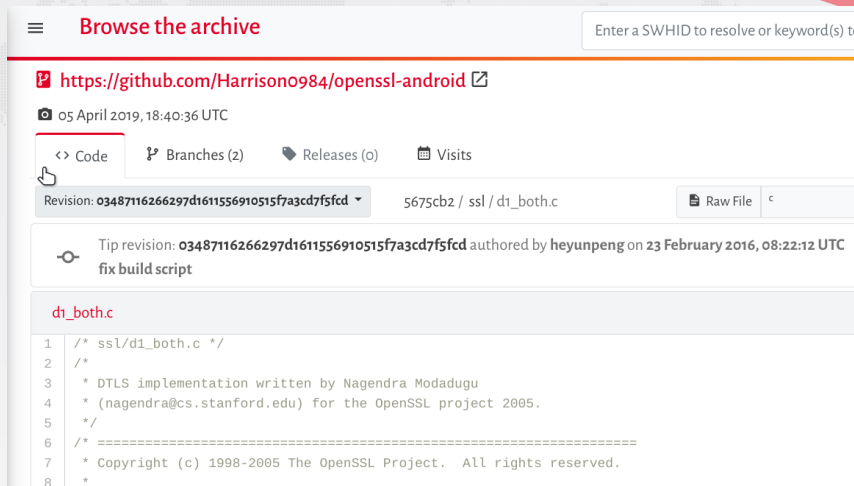
```
$ getrevdate.py --swhid swh:1:rev:03487116266297d1611556910515f7a3cd7f5fcd  
{'revision': {'swhid': 'swh:1:rev:03487116266297d1611556910515f7a3cd7f5fcd',  
  'date': {'date': '2016-02-23T16:22:12+08:00'}}}
```

Let's see where it comes from

```
$ grep "swh:1:rev:03487116266297d1611556910515f7a3cd7f5fcd" allrevs  
swh:1:cnt:de8bab873f2cf114d0d1b3e49acfa09bb9d0e4f7;path=ssl/d1_both.c;  
anchor=swh:1:rev:03487116266297d1611556910515f7a3cd7f5fcd;  
visit=swh:1:snp:7104ce60b2fa0650fe993195396a68f17dce5220;  
origin=https://github.com/Harrison0984/openssl-android
```

... more closely, cont'd

let's check [that occurrence](#)



≡ Browse the archive

<https://github.com/Harrisono984/openssl-android>

📷 05 April 2019, 18:40:36 UTC

[Code](#) [Branches \(2\)](#) [Releases \(0\)](#) [Visits](#)

Revision: **03487116266297d1611556910515f7a3cd7f5fcd** 5675cb2 / ssl / d1_both.c [Raw File](#)

Tip revision: **03487116266297d1611556910515f7a3cd7f5fcd** authored by heyunpeng on 23 February 2016, 08:22:12 UTC
fix build script

d1_both.c

```
1  /* ssl/d1_both.c */
2  /*
3   * DTLS implementation written by Nagendra Modadugu
4   * (nagendra@cs.stanford.edu) for the OpenSSL project 2005.
5   */
6  /* =====
7   * Copyright (c) 1998-2005 The OpenSSL Project. All rights reserved.
8   *
```

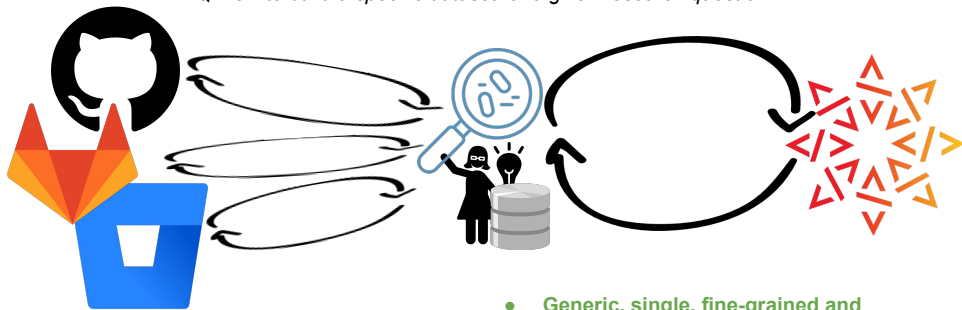
- 
- 1 Introduction
 - 2 The (open source) software tidal wave
 - 3 (Open Source) Software Supply Chain
 - 4 Meet Software Heritage
 - 5 Meet the Software Heritage dataset(s)
 - 6 Efficient traversal of the full graph
 - 7 Demo time!
 - 8 Impact on ESE studies**
 - 9 Call to action

Selected research works using Software Heritage

- 
-  Thibault Allançon, Antoine Pietri, Stefano Zacchioli
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 Zacchioli
Gender Differences in Public Code Contributions: a 50-year Perspective
IEEE Softw. 38(2): 45-50 (2021)
 -  Antoine Pietri, Guillaume Rousseau, Stefano Zacchioli
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 Zacchioli
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 Zacchioli
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 Zacchioli
Software Provenance Tracking at the Scale of Public Source Code
Empirical Software Engineering 25(4): 2930-2959 (2020)

Mining Android Applications on Software Heritage

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



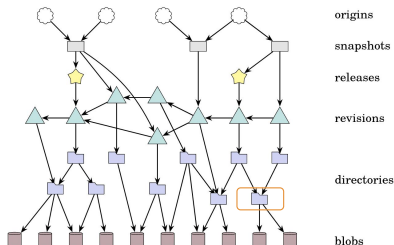
- **Specific and limited API**
- **Hardly reproducible**

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

(from the Inria/IRISA DiverSE team)

Using the SWH merkle dag to identify android repositories

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

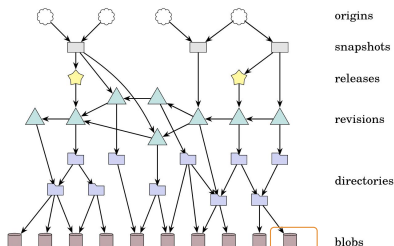


SWH Merkle DAG, Antoine Pietri

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

Using the SWH merkle dag to identify android repositories

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

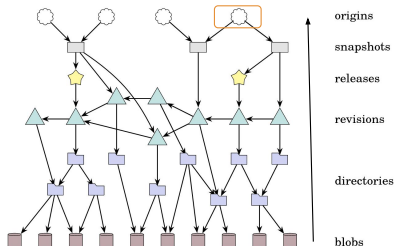


SWH Merkle DAG, Antoine Pietri

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

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

Broad variety of sources in *one open dataset*

reduces usual GH bias

Reference simple *standard data format*

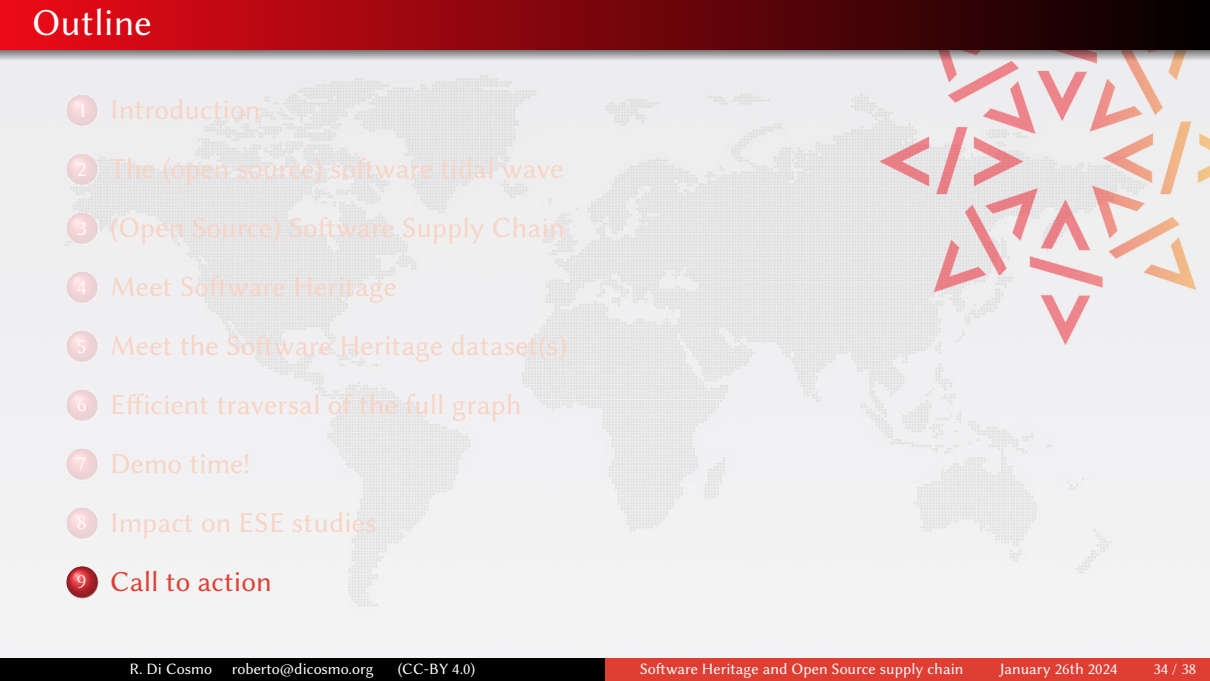
VCS and forge details are abstracted away

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

- 
- 1 Introduction
 - 2 The (open source) software tidal wave
 - 3 (Open Source) Software Supply Chain
 - 4 Meet Software Heritage
 - 5 Meet the Software Heritage dataset(s)
 - 6 Efficient traversal of the full graph
 - 7 Demo time!
 - 8 Impact on ESE studies
 - 9 Call to action

Join a growing, active community

Core Team



All together, 2023 UNESCO Symposium



The first five years in just five minutes



Ambassadors, news, blog, media

- meet the [ambassadors](#)
- subscribe to the [newsletter](#)
- read the [blog](#)
- follow [@swheritage](#)

Adopt and share best practices for ARDC

Archiving and referencing

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

- archive and reference in Software Heritage (see [detailed HOWTO](#))

Describing and Citing/Crediting

For **software one wants to put forward**, add these **extra steps**:

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

We can (and must)

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

Policy remarks on the road ahead

Infrastructures for Software: avoid balkanisation, mutualise cost

- build on common, shared, open, non profit infrastructures
- join Software Heritage

development member/sponsor, mirror, contributor

adoption ambassador, learned societies, policy

research address the many scientific challenges

Walking the talk in Europe

ongoing full workpackage in FAIRCORE4EOSC interconnects infrastructures with Software Heritage

open now CHIST-ERA joint ORD call

deadline: 14/12/2022

Belgium, Czech Republic, France, Lithuania, Luxembourg, Poland, Slovakia, Switzerland, Turkey

"Processes and tools to describe, share, reference and archive software [...] that leverage existing initiatives, such as Software Heritage"

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!

use, disseminate, contribute, build&adapt research tools, ...

Let's work together!

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](#))