Securing the (Open Source) Software Supply chain

challenges and opportunities

Roberto Di Cosmo

Director, Software Heritage
Inria and Université de Paris Cité

2025

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline

Open Science

Short Bio: Roberto Di Cosmo

Computer Science professor in Paris, now working at INRIA

- *35+ years* of research (Theor. CS, Programming, Software Engineering, Erdos #: 3)
- *25+ years* of Free and Open Source Software
- *15+ 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)

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

                CAF     CODE500         # ASTRONAUT:    PLEASE CRANK THE
                TC      BANKCALL        #               SILLY THING AROUND
                CADR    GOPERF1
                TCF     GOTOPOOH        # TERMINATE
                TCF     P63SPOT3        # PROCEED       SEE IF HE'S LYING

P63SPOT4        TC      BANKCALL        # ENTER         INITIALIZE LANDING RADAR
                CADR    SETPOS1

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

Quake III source code (excerpt)

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

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

    return y;
}
```

Len Shustek, Computer History Museum 2006

"Source code provides a view into the mind of the designer."

Software 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 Andreesen, 2011

Technology

Software Defined Everything

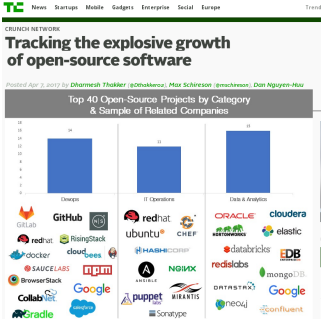
Hardware gets commoditised

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!*



News Startups Mobile Gadgets Enterprise Social Europe Trend

CRUNCH NETWORK

Tracking the explosive growth of open-source software

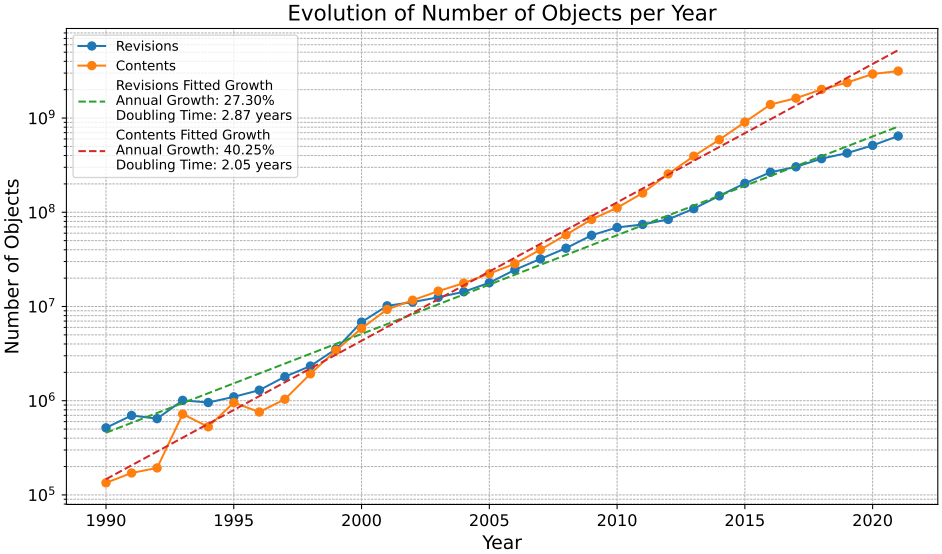
Posted Apr 7, 2017 by Dharmesh Thakker (@Dthakker02), Max Schireson (@mschireson), Dan Nguyen-Huu

Top 40 Open-Source Projects by Category & Sample of Related Companies

Reuse is the new rule

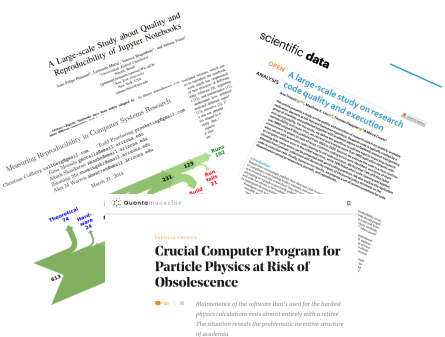
80% to 90% of a new application is … just reuse! (Sonatype survey, 2017)

(Open) Source Code grows at an exponential rate



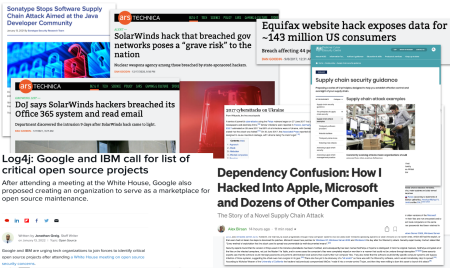
How are we managing our (open source) software ?

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

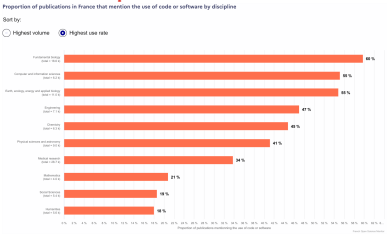
awareness is raising at the level of public policy

Outline

Open Science

Software Source Code: pillar of Open Science

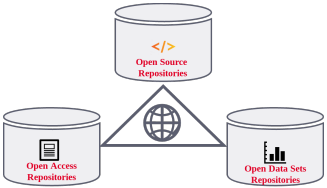
Software powers modern research



Over 20% of articles using software across all disciplines share it

2024 French Open Science Monitor

Key pillar: software



Links are important

Nota Bene

software may be a *tool*, a *research outcome* and a *research object*

access to the *source code* is essential!

A plurality of needs

Archive

Research software artifacts must be properly archived

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

Reference

Research software artifacts must be properly referenced

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

Describe

Research software artifacts must be properly described

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

Cite/Credit

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

to give *credit* to authors (*evaluation*!)

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

Open Science

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

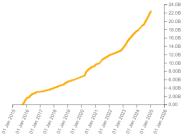
enable analysis of all software source code

A *universal* archive as a shared infrastructure

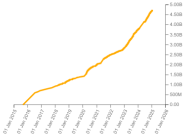
One infrastructure
open and shared

The largest archive ever built

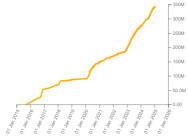
Source files
22,548,654,226



Commits
4,743,656,085



Projects
344,672,354



Directories
17,848,569,305

Authors
86,143,084

Releases
102,056,508

Bitbucket 2,578,475 origins	56,975 origins	git 33,350 origins
R 27,377 origins	debian 141,834 origins	88,561 origins
GitHub 249,868,189 origins	gitiles 24,378 origins	GitLab 5,736,223 origins
git 3,791 origins	Gogs 394 origins	GO 1,887,337 origins
Guix 68,391 origins	GNU 354 origins	heptapod 1,340 origins
launchpad 654,755 origins	Maven 312,179 origins	NixOS 48,905 origins
npm 4,003,267 origins	5,438 origins	Packagist 376,882 origins

An international, non profit initiative built for the long term

Sharing the vision

United Nations Educational, Scientific and Cultural Organization

Donors, members, sponsors

Inria — INVENTEURS DU MONDE NUMÉRIQUE

Diamond sponsors

cea, IBM, Microsoft

Platinum sponsors

cnrs, Ministère de l'enseignement supérieur et de la recherche, HUAWEI

Gold sponsors

amazon, Google, openinvention*network*, Université Paris Cité

Silver sponsors

AdaCore, Ministère des Armées, DGA, Premier Ministre, Direction interministérielle du numérique, GitHub, Sorbonne Université

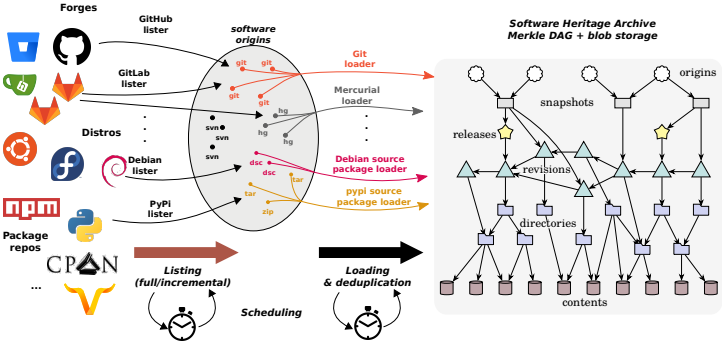
Bronze sponsors

cnes, SCANOSS, Scuola Normale Superiore, Sant'Anna, Université de Lille, Université de Liège, Università di Pisa

And many more ...

www.softwareheritage.org/support/testimonials

The archive, under the hood



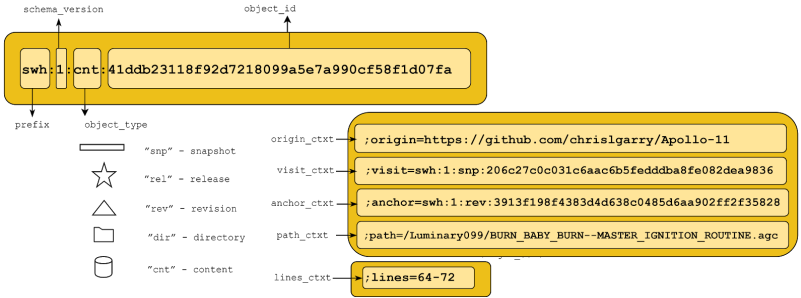
Global development history permanently archived in a uniform data model

- over 22 billion unique source files from over 340 million software projects
- ~2PB (compressed) blobs, ~50 B nodes, ~800 B edges

The Software Hash identifier (SWHID)

Software Heritage Identifiers (SWHID)

see swhid.org



50+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

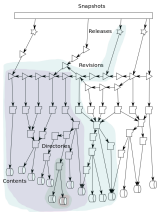
Examples: Apollo 11 AGC excerpt, Quake III rsqrt
Guidelines available, see the HOWTO

Breaking news: standardisation, see swhid.org - DIS 18670

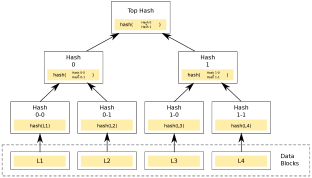
A revolutionary infrastructure

Modern "Library of Alexandria", *international, non profit, long term* initiative
addressing the needs of *industry, research, culture and society as a whole*

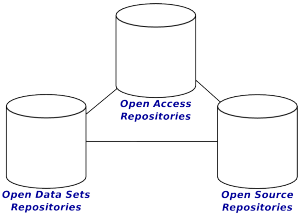
Software Graph



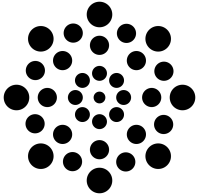
Software Blockchain



Open Science pillar



Big Code



One infrastructure, *shared*: more efficient, less waste …

… supporting a broad range of applications

Outline

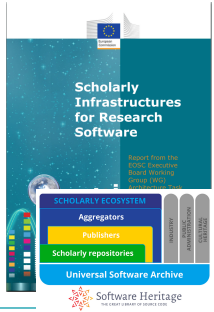
A walkthrough

- Browse (e.g. Apollo 11 [excerpt], your work may be already there !)
- Trigger archival, use the updateswh browser extension, configure the webhooks
- Get and use SWHIDs (full specification available online)
- Cite software with biblatex-software package from CTAN
  - Overleaf ACMART template available
- Example in journals: article from IPOL
- Example with Parmap: devel on Github, archive in SWH, curated deposit in HAL
- Extracting all the software products for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools
- Curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, ...
- Example use in research articles:
  - compare Fig. 1 and conclusions in the 2012 version and the updated version
  - SWHID in a replication experiment

Outline

A few adoption indicators

Policy



- Recommendations in ANR 2023 guidelines (p. 17)
- HAL+SWH in the Open Science software booklet

Projects

eosc | FAIRCORE4EOSC
Core Components Supporting a FAIR EOSC

The CodeMeta Project

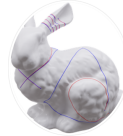
eosc | FAIR-IMPACT
Expanding FAIR solutions across EOSC

Users and collaborations

What are they "referencing"?

source	n	percentage
Not available	2868	46.22
GitHub	1151	18.55
software heritage	387	6.24
zenodo	142	2.29
r package	70	1.13
cran	56	0.90
r package version	54	0.87
gitlab	35	0.56

Graphics Replicability Stamp Initiative



b/Surf: Interactive Bézier Splines on Surface Meshes

Claudio Mancinelli, Giacomo Nazzaro, Fabio Pellacini, Enrico Puppo
IEEE Transactions on Visualization and Computer Graphics (TVCG)

doi | Repository

Outline

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.

Contact

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-inventory/`

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;
```

Results

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

Results (20) | Copy | Download results

Search rows | 1

#	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

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 (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~~ Rust and gRPC APIs available …

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

Outline

Policy highlights: industry and sovereignty

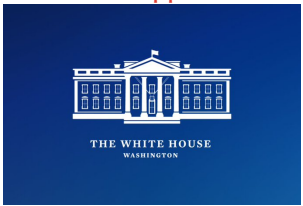
Like KYC in banking, KYSW is now essential all over IT…

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

Open source security

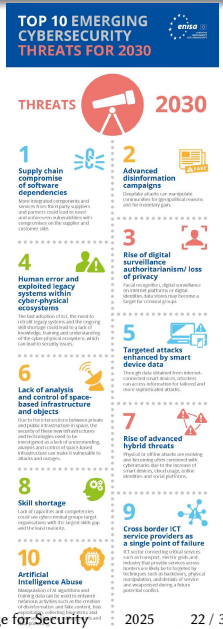
Open source software can be freely used, copied, and modified.

Open Source Software (OSS) is everywhere

- Huge boost for innovation! (e.g., reduced time to market)
- 96% of (non-open) software products depend on open source (2022).
- Open source is at the heart of the global digital infrastructure.

With great exposure comes great scrutiny…

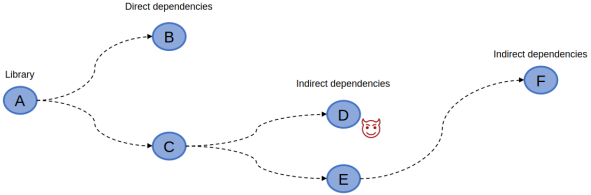
- …by both good and bad actors.
- OSS is more and more targeted by attackers.
- Increased policy attention to secure OSS, e.g.:
  - US: Biden's executive orders (2022, Jan 2025!)
  - EU: CRA, progressively coming into effect

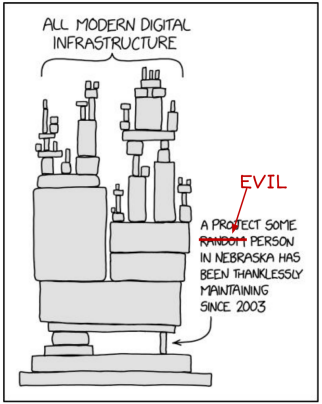


Software supply chain attacks

Reusing OSS via dependencies

- Software dependencies: a popular way of reusing open source software.
- Software product A uses functionalities implemented in OSS product B ... and so on.





based on xkcd.com/2347

Attacking the software supply chain

- Attacking undermaintained "leaf" packages (e.g., D) → efficient attack strategy
- Many documented attacks: event-stream (2018), node-ipc (2022), XZ utils (2024), ...

Securing open source with Software Heritage

What does Software Heritage bring to the table?

- The largest archive that guarantees the:
  1. availability,
  2. integrity, and
  3. traceability of (OSS) source code.
- A universal, open knowledge base of *facts* about open source software…
- …that can be leveraged by everyone (not only the big players) to secure OSS.

SWHSec project

swhsec.github.io

- 2023–2027 R&D project, funded by French national CampusCyber
- 8 research teams, including SWH core

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

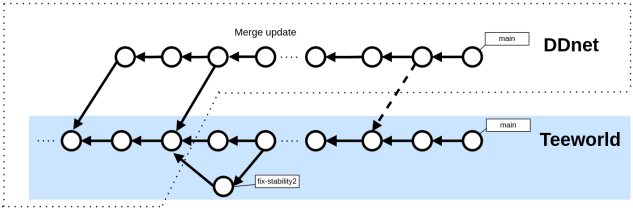
Axes: (1) extending SWH with security info + (2) code analysis, dependency analysis, vulnerability tracking, automatic vulnerability fixing, … at SWH scale.

One-day vulnerabilities

- Def.: vulnerabilities that are publicly known, but not fixed yet in software you use.
- Challenge: identify them quickly and exhaustively, then apply countermeasures.
- Many tools available to detect one-day vulnerabilities *via declared dependencies*.

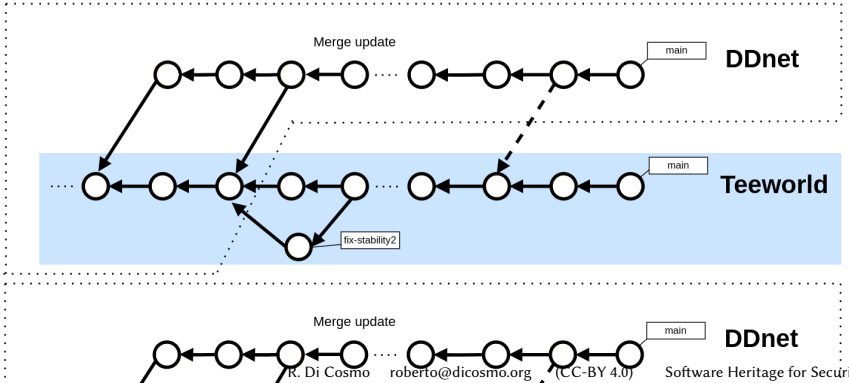
Reusing OSS via forks

Open source is also reused via forking: (1) start from existing OSS (e.g., Teeworlds game), (2) create your own (e.g., DDnet), (3) periodically integrate changes.



Vulnerability propagation through forks

- Any change to a piece of software (*commit*) can introduce a new vulnerability.
- Or it can fix an existing vulnerability.
- What happens if a project is forked between introduction and fix of a vulnerability?
- It inherits the vulnerability, ... until the change with the fix is integrated.



swh-vuln: chasing one-day vulnerabilities across forks… at SWH scale

Approach

1. Start from a public DB of vuln. introduced/fixed in public commits (e.g., OSV.dev).
2. "Color" the entire graph of public code development history with vulnerability info.
   - Software Heritage is the only place where this can be done at the scale of all forks, across all public code, independently of specific development platforms.
3. Inform maintainers of vulnerable forks. (After validation.)

Early results

- Identified 2.2 M (million) forks of repositories referenced by OSV.dev, containing vulnerable commits; 1.3 M forks vulnerable in their most recent commit.
- 86.6 M vulnerable commits were specific to forks, not findable with current tools.
- Among 66 manually vetted cases, 5 confirmed vulnerabilities (1 critical).

Romain Lefeuvre, Charly Reux, Stefano Zacchiroli, Olivier Barais, Benoit Combemale
Chasing One-day Vulnerabilities Across Open Source Forks
To appear, 2025.

An universal knowledge base about public code vulnerabilities

Vision

- Software Heritage is the perfect (and only) place where to build an universal knowledge base that maps known vulnerabilities to public code artifacts.
- SWH can provide an open data API mapping SWHIDs to CVEs, that knows about *all public commits* and can be leveraged to increase open source security.

EU Cyber Resilience Act (CRA)

- Key helper to abide to CRA obligations, coming into effect ~Q3 2026.
- SWH funding member of the Open Regulatory Compliance Working Group.

Roadmap

- Context: SWHSec project (PTCC-funded, 2023-2027).
- Current status: working prototype that processes OSV.dev data and use it to "color" the entire SWH commit graph (~5 billion commits) with vulnerability information.

Outline

Open Science

A growing and active community

Core Team

All together, 2024 Symposium

Ambassadors

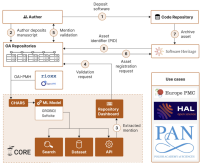
Becoming an Ambassador

Interested in becoming a Software Heritage ambassador?
Tell us about yourself and your interest in our mission.

ambassadorprogram@softwareheritage.org

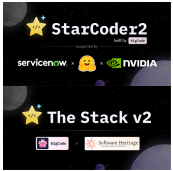
Software Heritage: 2024 progress hilights

January 2024 — SoFAIR Kickoff



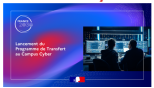
Massive identification, archival and reference of software in academic publications

February 2024 — First AI contact



StarCoder2 trained on GitHub subset of Software Heritage

February 2024 — SWHSec Kickoff



SWHSec: 5 years PTCC project

Software supply chain Security

Academic interoperability

eosc

FAIR-IMPACT

FAIRCORE4EOSC

Connect with Zenodo, Episcience, Dagsthul, SwMath, OpenAire, …

Fall 2024 — CodeCommons

Building the reference software knowledge base for AI

… and much more

2024 annual report here →

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, ...