Software Heritage

Large-Scale Research on Public Code Development

Stefano Zacchiroli

Télécom Paris, Institut Polytechnique de Paris
stefano.zacchiroli@telecom-paris.fr

4 Feb 2026 — ENS, Lyon

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

About the speaker

- Professor of Computer Science, Télécom Paris, Institut Polytechnique de Paris
- Free/Open Source Software activist (20+ years)
- Debian Developer & Former 3x Debian Project Leader
- Former Open Source Initiative (OSI) director
- Software Heritage co-founder & Chief Scientific Officer (CSO)

Outline

1. Software as knowledge
2. Software Heritage
3. Research highlights: datasets for the masses
4. Research highlights: software evolution
5. Research highlights: diversity, equity, and inclusion
6. Research highlights: cybersecurity
7. Conclusion

Software is dual-form knowledge

> *“The source code for a work means the preferred form of the work for making modifications to it."*
>
> *GPL Licence*

Hello World

Program (excerpt of binary)

```
4004e6:  55
4004e7:  48 89 e5
4004ea:  bf 84 05 40 00
4004ef:  b8 00 00 00 00
4004f4:  e8 c7 fe ff ff
4004f9:  90
4004fa:  5d
4004fb:  c3
```

Program (source code)

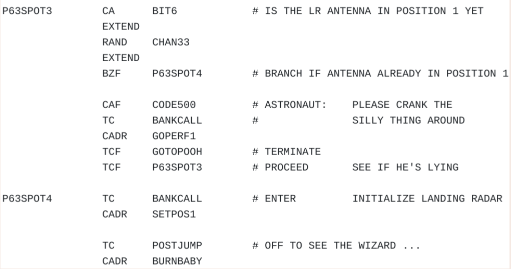
```
/* Hello World program */

#include<stdio.h>

void main()
{
    printf("Hello World");
}
```

Software *source code* is precious human knowledge

Harold Abelson, Structure and Interpretation of Computer Programs (1st ed.) 1985

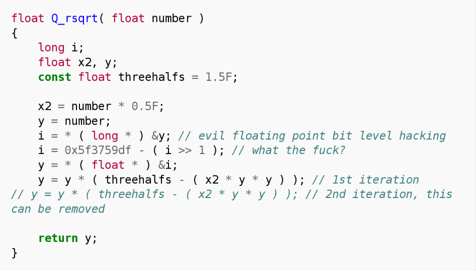
“Programs must be written for people to read, and only incidentally for machines to execute.”

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 source code is fragile

Like all digital information, FOSS is fragile

- link rot: projects are created, moved around, removed
- business-driven code loss (e.g., Gitorious, Google Code, Bitbucket)
- data rot: physical media with legacy software decay

If a website disappears you go to the Internet Archive…

where do you go if (a repository on) GitHub or GitLab goes away?

Outline

1. Software as knowledge
2. Software Heritage
3. Research highlights: datasets for the masses
4. Research highlights: software evolution
5. Research highlights: diversity, equity, and inclusion
6. Research highlights: cybersecurity
7. Conclusion

Software Heritage, in a nutshell

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

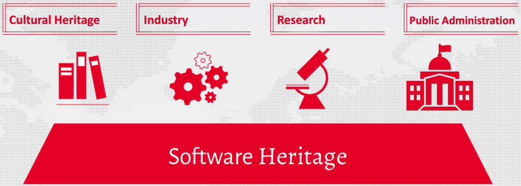
preserve and share all software source code

Research infrastructure

enable analysis of all software source code

The largest software archive, a shared infrastructure

One infrastructure
open and shared



Cultural Heritage

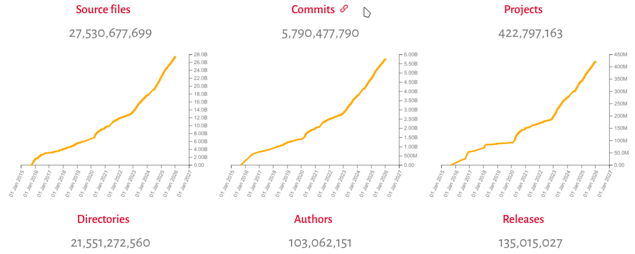
Industry

Research

Public Administration

Software Heritage

The largest archive ever built



Source files
27,530,677,699

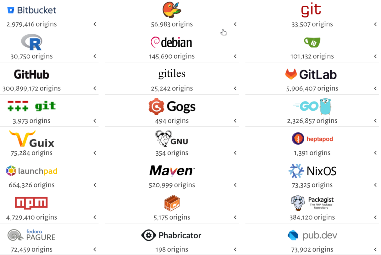
Commits
5,790,477,790

Projects
422,797,163

Directories
21,551,272,560

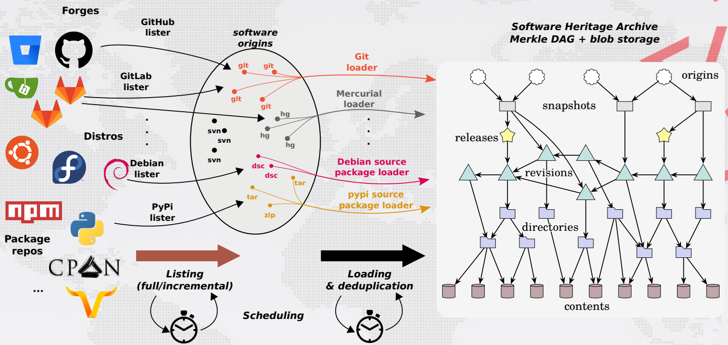
Authors
103,062,151

Releases
135,015,027



Bitbucket 2,979,416 origins	56,983 origins	git 33,507 origins
R 30,750 origins	debian 145,690 origins	101,132 origins
GitHub 300,899,172 origins	gitiles 25,242 origins	GitLab 5,906,407 origins
+++ git 3,973 origins	Gogs 494 origins	GO 2,326,857 origins
Guix 75,284 origins	GNU 354 origins	heptapod 1,391 origins
launchpad 664,326 origins	Maven 520,999 origins	NixOS 73,325 origins
npm 4,729,410 origins	5,175 origins	Packagist 384,120 origins
fedora PAGURE 72,459 origins	Phabricator 198 origins	pub.dev 73,902 origins

A peek under the hood: a universal archive

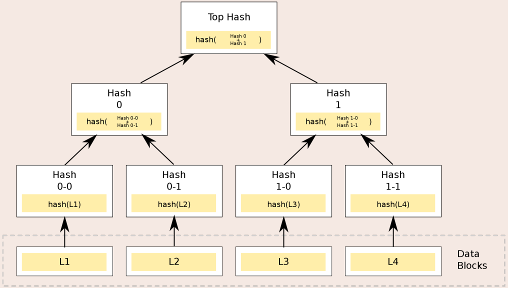


Global development history permanently archived in a uniform data model

- over 27 billion unique source files from over 420 million software projects
- ~2PB (compressed) blobs, ~50 B nodes, ~1 trillion edges

Merkle trees

Merkle tree (R. C. Merkle, CRYPTO 1987)



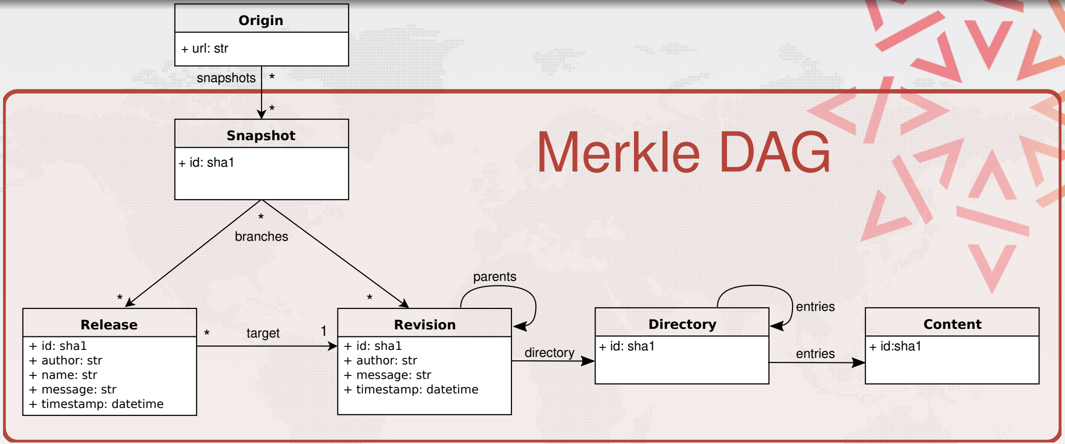
Combination of

- tree
- hash function

Classical cryptographic construction

- fast, parallel signature of large data structures
- widely used (e.g., Git, blockchains, IPFS, ...)
- built-in deduplication

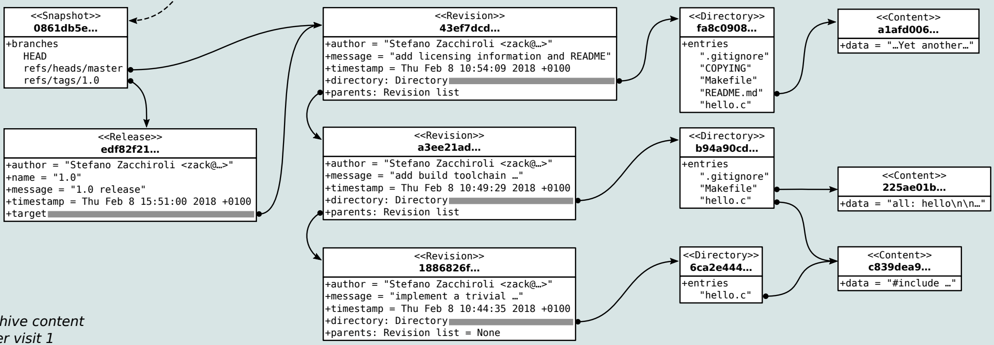
Data model



A global graph linking together fully deduplicated source code artifact (files, commits, directories, releases, etc.) to the places that distribute them (e.g., Git repositories), providing a unified view on the entire *Software Commons*.

The archive: a (giant) Merkle DAG

origin	*visit*	*snapshot*	*timestamp*
https://forge.softwareheritage.org/source/helloworld.git	1	0861db5e...	Fri Feb 9 12:38:45 2018 +0100



```
<<Snapshot>>
0861db5e...
+branches
   HEAD
   refs/heads/master
   refs/tags/1.0

<<Release>>
edf82f21...
+author = "Stefano Zacchiroli <zack@…>"
+name = "1.0"
+message = "1.0 release"
+timestamp = Thu Feb 8 15:51:00 2018 +0100
+target

<<Revision>>
43ef7dcd...
+author = "Stefano Zacchiroli <zack@…>"
+message = "add licensing information and README"
+timestamp = Thu Feb 8 10:54:09 2018 +0100
+directory: Directory
+parents: Revision list

<<Revision>>
a3ee21ad...
+author = "Stefano Zacchiroli <zack@…>"
+message = "add build toolchain …"
+timestamp = Thu Feb 8 10:49:29 2018 +0100
+directory: Directory
+parents: Revision list

<<Revision>>
1886826f...
+author = "Stefano Zacchiroli <zack@…>"
+message = "implement a trivial …"
+timestamp = Thu Feb 8 10:44:35 2018 +0100
+directory: Directory
+parents: Revision list = None

<<Directory>>
fa8c0908...
+entries
   ".gitignore"
   "COPYING"
   "Makefile"
   "README.md"
   "hello.c"

<<Directory>>
b94a90cd...
+entries
   ".gitignore"
   "Makefile"
   "hello.c"

<<Directory>>
6ca2e444...
+entries
   "hello.c"

<<Content>>
a1afd006...
+data = "…Yet another…"

<<Content>>
225ae01b...
+data = "all: hello\n\n…"

<<Content>>
c839dea9...
+data = "#include …"
```

Archive content after visit 1

origin	*visit*	*snapshot*	*timestamp*

Demo time!

- Browse the archive
- Trigger archival of your preferred software in a breeze
- Get and use SWHIDs (full specification available online)
- The Apollo 11 AGC source code example
- Cite software with the biblatex-software style from CTAN
- Example use in a research article: compare Fig. 1 and conclusions
  - in the 2012 version
  - in the updated version using SWHIDs and Software Heritage
- Example in a journal: an article from IPOL
- Curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, . . .
- Rescue landmark legacy software, see the SWHAP process with UNESCO

Outline

1. Software as knowledge
2. Software Heritage
3. Research highlights: datasets for the masses
4. Research highlights: software evolution
5. Research highlights: diversity, equity, and inclusion
6. Research highlights: cybersecurity
7. Conclusion

Software Heritage dataset portal

datasets.softwareheritage.org

Graph dataset

Use case: large scale analyses of the most comprehensive corpus on the development history of free/open source software.

Antoine Pietri, Diomidis Spinellis, Stefano Zacchiroli
The Software Heritage Graph Dataset: Public software development under one roof
MSR 2019: 16th Intl. Conf. on Mining Software Repositories. IEEE
preprint: http://deb.li/swhmsr19

Dataset

- Relational representation of the full graph as a set of tables
- Available as open data: docs.softwareheritage.org/devel/swh-dataset/graph
- Chosen as subject for the MSR 2020 Mining Challenge

Formats

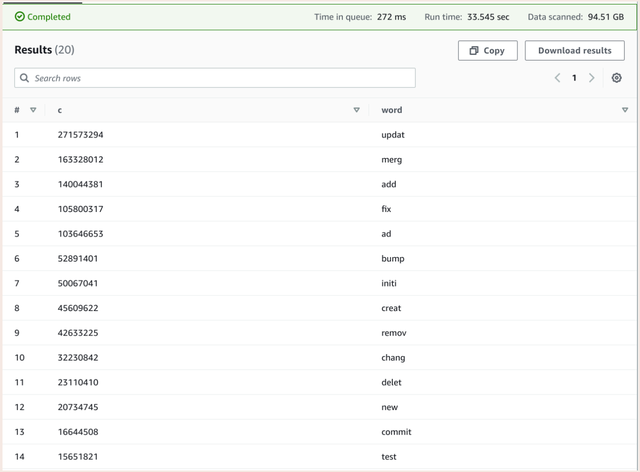
- Local use: set of Apache ORC files (10+ TiB in total)
- Live usage: Amazon Athena (SQL-queriable), Azure Data Lake

Graph dataset — example

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)

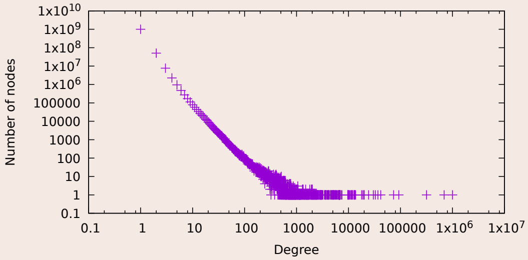
#	c	word
1	271573294	updat
2	163328012	merg
3	140044381	add
4	105800317	fix
5	103646653	ad
6	52891401	bump
7	50067041	initi
8	45609622	creat
9	42633225	remov
10	32230842	chang
11	23110410	delet
12	20734745	new
13	16644508	commit
14	15651821	test

Graph dataset — example

Fork arity

i.e., how often is a commit based upon?

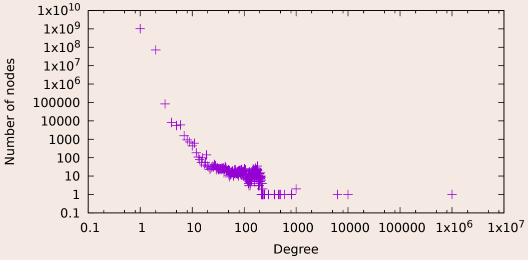
```
SELECT fork_deg, count(*) FROM (
  SELECT id, count(*) AS fork_deg
  FROM revision_history GROUP BY id) t
GROUP BY fork_deg ORDER BY fork_deg;
```



Merge arity

i.e., how large are merges?

```
SELECT merge_deg, COUNT(*) FROM (
  SELECT parent_id, COUNT(*) AS merge_deg
  FROM revision_history GROUP BY parent_id) t
GROUP BY merge_deg ORDER BY merge_deg;
```



License dataset

Stefano Zacchiroli
A Large-scale Dataset of (Open Source) License Text Variants
MSR 2022 (best dataset paper) + Empir. Soft. Eng. 28(6): 147 (2023)
preprint: `https://arxiv.org/abs/2308.11258`

Dataset

- 6.9 million unique full texts of FOSS license variants
- Detected using filename patterns across the entire SWH archive
  - `LICENSE`, `COPYRIGHT`, `NOTICE`, etc.
- Metadata: file lengths measures, detected MIME type, detected SPDX license (via ScanCode), example origin repository, oldest public commit of origin, ground truth

Use cases

- Empirical studies on FOSS licensing, including phylogenetics
- Training of automated license classifiers
- NLP analyses of legal texts

The Software Heritage Filesystem (SwhFS)

The Software Heritage Filesystem (SwhFS) is a user-space POSIX filesystem that enables browsing parts of the Software Heritage archive as if it were locally available.

- Code: forge.softwareheritage.org/source/swh-fuse
- Documentation: docs.softwareheritage.org/devel/swh-fuse

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

The Software Heritage Filesystem (SwhFS) — example

```
$ mkdir swhfs
$ swh fs mount swhfs/  # mount the archive
$ cd swhfs/

$ cat archive/swh:1:cnt:c839dea9e8e6f0528b468214348fee8669b305b2
#include <stdio.h>

int main(void) {
    printf("Hello, World!\n");
}

$ cd archive/swh:1:dir:1fee702c7e6d14395bbf5ac3598e73bcbf97b030
$ ls | wc -l
127
$ grep -i antenna THE_LUNAR_LANDING.s | cut -f 5
# IS THE LR ANTENNA IN POSITION 1 YET
# BRANCH IF ANTENNA ALREADY IN POSITION 1
```

The Software Heritage Filesystem (SwhFS) — example (cont.)

```
$ cd archive/swh:1:rev:9d76c0b163675505d1a901e5fe5249a2c55609bc

$ ls -F
history/  meta.json@  parent@  parents/  root@

$ jq ’.author.name, .date, .message’ meta.json
"Michal Golebiowski-Owczarek"
"2020-03-02T23:02:42+01:00"
"Data:Event:Manipulation: Prevent collisions with Object.prototype ..."

$ find root/src/ -type f -name ’*.js’ | xargs cat | wc -l
10136
```

Graph compression

Q: Is it possible to efficiently perform software development history analyses at the scale of Software Heritage archive on a single, relatively cheap machine?

Idea

Apply graph compression techniques from the field of network analysis.

Results

The entire archive graph (35 B nodes, 500 B edges) can be loaded in 300 GiB and then traversed at the cost of tens of ns per edge (= a few hours for a full single-thread visit).

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

Tommaso Fontana, Sebastiano Vigna, Stefano Zacchiroli
WebGraph: The Next Generation (Is in Rust)
WWW'24, the ACM Web Conference 2024

Rust and gRPC APIs available: docs.softwareheritage.org/devel/swh-graph/

Background — (Web) graph compression

Definition (The graph of the Web)

Directed graph that has Web pages as nodes and hyperlinks between them as edges.

Properties (1)

- **Locality:** pages link to pages whose URLs are lexicographically similar. URLs share long common prefixes.

→ use D-gap compression

Adjacency lists

Node	Outdegree	Successors
…	…	…
15	11	13,15,16,17,18,19,23,24,203,315,1034
16	10	15,16,17,22,23,24,315,316,317,3041
17	0	
18	5	13,15,16,17,50
…	…	…

D-gapped adjacency lists

Node	Outdegree	Successors
…	…	…
15	11	3,1,0,0,0,0,3,0,178,111,718
16	10	1,0,0,4,0,0,290,0,0,2723
17	0	
18	5	9,1,0,0,32
…	…	…

Background — (Web) graph compression (cont.)

Definition (The graph of the Web)

Directed graph that has Web pages as nodes and hyperlinks between them as edges.

Properties (2)

- Similarity: pages that are close together in lexicographic order tend to have many common successors.

→ use reference compression

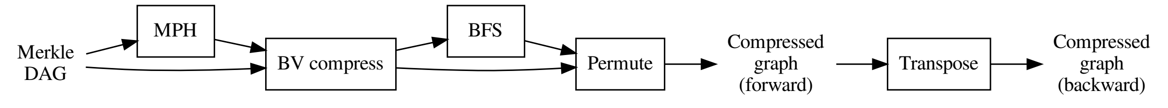
Adjacency lists

Node	Outd.	Successors
...	...	...
15	11	13,15,16,17,18,19,23,24,203,315,1034
16	10	15,16,17,22,23,24,315,316,317,3041
17	0	
18	5	13,15,16,17,50
...	...	...

Copy lists

Node	Ref.	Copy list	Extra nodes
...	...	...	...
15	0		13,15,16,17,18,19,23,24,203,315,1034
16	1	01110011010	22,316,317,3041
17			
18	3	11110000000	50
...	...	...	

Graph compression pipeline



- MPH: minimal perfect hash, mapping Merkle IDs to 0..N-1 integers
- BV compress: Boldi-Vigna compression (based on MPH order)
- BFS: breadth-first visit to renumber
- Permute: update BV compression according to BFS order

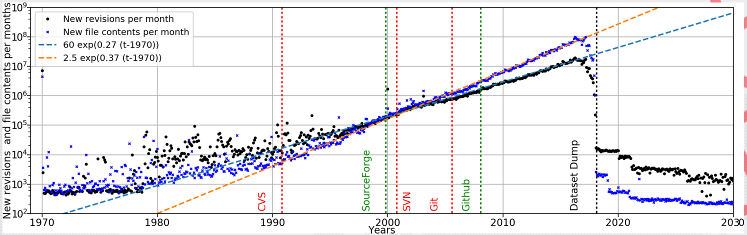
(Re)establishing locality

- Key for good compression is a node ordering that ensures locality and similarity
- Which is very much *not* the case with Merkle IDs, …but is the case *again* after BFS(+LLP) reordering

Outline

1. Software as knowledge
2. Software Heritage
3. Research highlights: datasets for the masses
4. Research highlights: software evolution
5. Research highlights: diversity, equity, and inclusion
6. Research highlights: cybersecurity
7. Conclusion

Software provenance and evolution



Key findings

- The amount of original commits in public code doubles every ~30 months and has been doing so for 20+ years; original source code files double every ~22 months
- It is possible to trace the provenance of source code artifacts at this scale in a compact relational model via the notion of isochrone graphs.

Rousseau, Di Cosmo, Zacchiroli

Software Provenance Tracking at the Scale of Public Source Code

Empir. Softw. Eng. 25(4): 2930-2959 (2020)

Outline

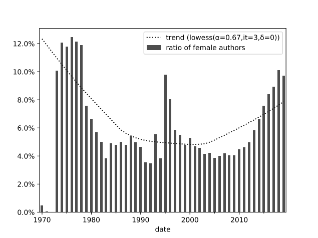
1. Software as knowledge
2. Software Heritage
3. Research highlights: datasets for the masses
4. Research highlights: software evolution
5. Research highlights: diversity, equity, and inclusion
6. Research highlights: cybersecurity
7. Conclusion

Diversity, equity, and inclusion

Archived commit metadata contains public information that can be mined to study DEI traits of the global population of public code authors.

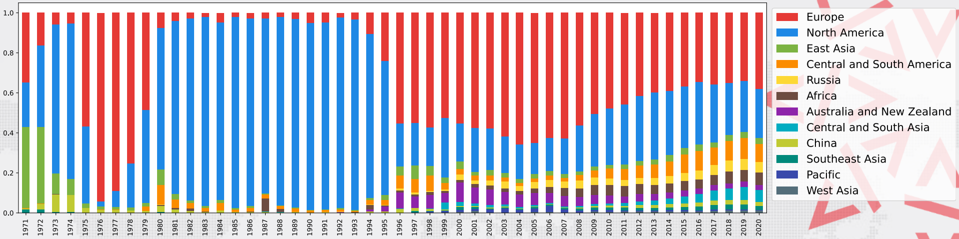
Gender gap — key findings

- Male authors contributed 92% of public code commits up to 2019.
- Female authors (and their commits) have grown stably for 15 years reaching 10% of yearly commits in 2019.
- The COVID-19 pandemic has *caused* a trend inversion (it is not just correlation!)



- Zacchiroli. *Gender differences in public code contributions: a 50-year perspective.* IEEE Software, 2021
- Rossi and Zacchiroli. *Worldwide gender differences in public code contributions [. . .].* ICSE SEIS, 2022
- Casanueva et al. *The Impact of the COVID-19 Pandemic on Women's Contribution to Public Code.* Empir. Softw. Eng. 30(25), 2025

Diversity, equity, and inclusion (cont.)



Geographic gap — key findings

- Early decades of public code dominated by contributions from North America, followed by a period of alternating dominance between North America and Europe.
- Since then geographic diversity has increased constantly, with raising importance of contributions from Central and South America.
- Geo *and* Gender gap: the trend of increased female contributions is global, with the exception of some regions in Asia where it is either slower or flat.

Rossi and Zacchiroli. *Geographic diversity in public code contributions.* MSR 2022

Outline

1. Software as knowledge
2. Software Heritage
3. Research highlights: datasets for the masses
4. Research highlights: software evolution
5. Research highlights: diversity, equity, and inclusion
6. Research highlights: cybersecurity
7. Conclusion

Open source security

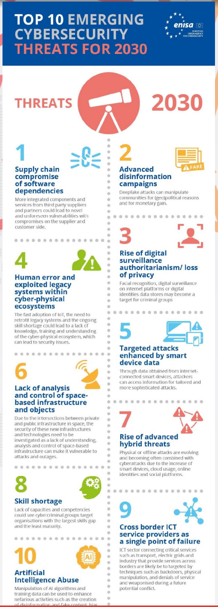
Open source software can be freely used, copied, and modified.

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



TOP 10 EMERGING CYBERSECURITY THREATS FOR 2030

THREATS 2030

1 **Supply chain compromise of software dependencies**
More integrated components and services from third party suppliers and partners could lead to novel and unforeseen vulnerabilities with compromises on the supplier and customer side.

2 **Advanced disinformation campaigns**
Deepfake attacks can manipulate communities for (geo)political reasons and for monetary gain.

3 **Rise of digital surveillance authoritarianism/ loss of privacy**
Facial recognition, digital surveillance on internet platforms or digital identities data stores may become a target for criminal groups

4 **Human error and exploited legacy systems within cyber-physical ecosystems**
The fast adoption of IoT, the need to retrofit legacy systems and the ongoing skill shortage could lead to a lack of knowledge, training and understanding of the cyber-physical ecosystem, which can lead to security issues.

5 **Targeted attacks enhanced by smart device data**
Through data obtained from internet-connected smart devices, attackers can access information for tailored and more sophisticated attacks.

6 **Lack of analysis and control of space-based infrastructure and objects**
Due to the intersections between private and public infrastructure in space, the security of these new infrastructures and technologies need to be investigated as a lack of understanding, analysis and control of space-based infrastructure can make it vulnerable to attacks and outages.

7 **Rise of advanced hybrid threats**
Physical or offline attacks are evolving and becoming often combined with cyberattacks due to the increase of smart devices, cloud usage, online identities and social platforms.

8 **Skill shortage**
Lack of capacities and competencies could see cybercriminal groups target organisations with the largest skills gap and the least maturity.

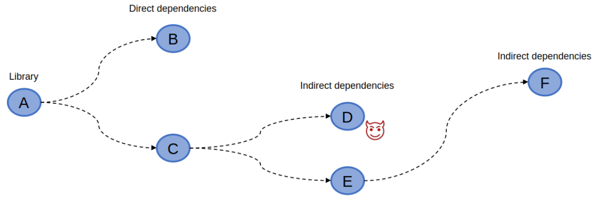
9 **Cross border ICT service providers as a single point of failure**
ICT sector connecting critical services such as transport, electric grids and industry that provide services across borders are likely be to targeted by techniques such as backdoors, physical manipulation, and denials of service and weaponised during a future potential conflict.

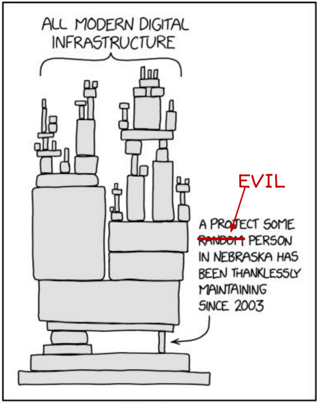
10 **Artificial Intelligence Abuse**
Manipulation of AI algorithms and training data can be used to enhance nefarious activities such as the creation of disinformation and fake content, bias

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.




ALL MODERN DIGITAL INFRASTRUCTURE
EVIL
A PROJECT SOME ~~RANDOM~~ PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

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), . . .

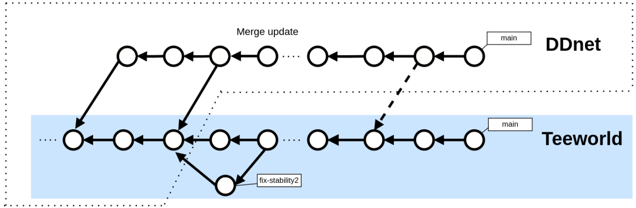
One-day vulnerabilities in open source

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.

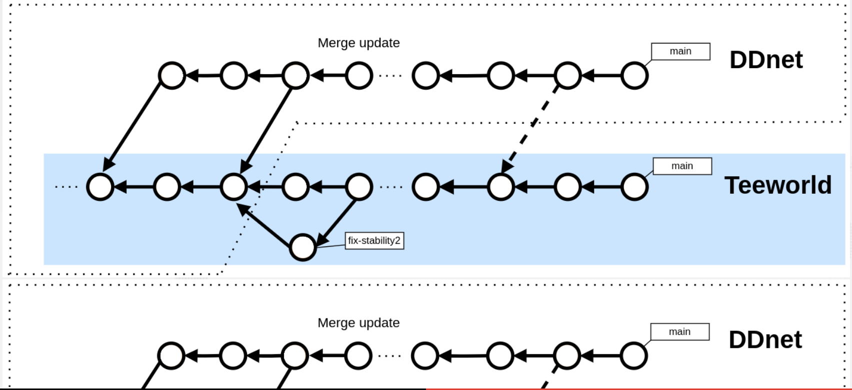
Reusing OSS via forks

…but OSS 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.)

Results

- Starting from 7162 repos in OSV, we identified 1.7 M forks potentially vulnerable in their most recent commit.
  - 86.6 M vulnerable commits were specific to forks, not findable with current tools.
- We manually verified 152 cases, confirming 135 high-severity vulnerabilities in popular forks; 9 were further confirmed by maintainers.

Romain Lefeuvre, Charly Reux, Stefano Zacchiroli, Olivier Barais, Benoit Combemale
Chasing One-day Vulnerabilities Across Open Source Forks
https://arxiv.org/abs/2511.05097, Nov 2025.

Git repository alterations

Git allows rewriting history (of the version control kind!)

```
$ git rebase --interactive <...>
$ git push --force
```

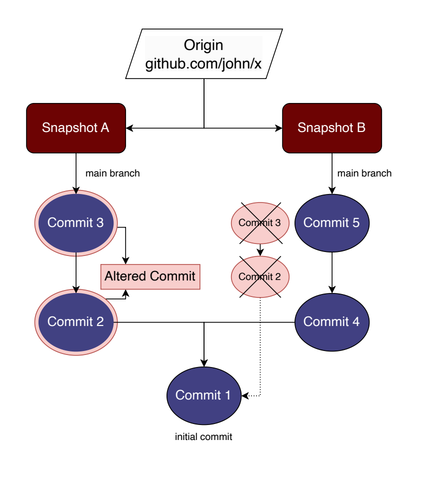
- Useful feature! E.g., to clean your code before sharing it
- Also annoying and risky on public branches
  - Hinders reproducibility and voids availability of specific Git objects
  - Supply chain concerns: what was altered and why?

1. How often are public Git histories altered?
2. When this happens, what is changed and why?

Note: forges do not keep the history of these modifications. SWH is the only place where they can be analyzed.

A large-scale analysis of Git repository alterations

1. Retrieve from Software Heritage 111 M Git repositories (1) archived at least twice, (2) with different states ("snapshots")
2. For each repository, compare snapshots 2-by-2 to detect altered histories and the root cause of each alteration
3. Classify altered commits by what changed before/after alteration



Key findings on destructive repository alterations

How often?

- 1.22M repositories contain altered histories (~1.1%)
- 8.7M altered commits → Pro tip: make sure your important commits are in SWH!

Where?

- Pull request branches: 37.6% → Might be OK, dependening on workflow
- main/master branches: 11.4% → Concerning!

What?

- Commit metadata (13.3%): author, date, message, … → Risky for provenance & IP
- File/dir. changes (76.8%): *retroactive* file modifications and/or deletions

Case studies

Case study #1: License changes

- ~800K retroactively altered license files on main branches
- Spanning 32k repositories (76 with 1000+ stars)
- 79% version updates (e.g., GPL 2→3)
- 14% full changes (e.g., MIT→GPL)
- Serious concern: retroactive changes may *de facto* suppress previously granted rights (without an archival copy!)

Case study #2: Removing secrets

- 13M file removals involved files/paths referring to "secrets"
  - Examples: private keys, certificates, passwords
- Spanning 75k repositories
- Issue: History alteration $\not\Rightarrow$ security (archived copies persist)
- Keys must be rotated, not only purged from Git
- Might indicate poor security practices.

GitHistorian prototype

- Imagine you would like to avoid repositories with a track record of history alterations, or at least be alerted about them, for vetting purposes. How can you?
- For demonstration purposes only, we developed GitHistorian, a prototype OSS tool that leverages SWH data to address this need.

```
$ git-historian check https://github.com/example/project --branch main --verbose
Connected to the Software Heritage database!
Found 2 altered history records for 'https://github.com/example/project'

Record #1:
  Branch Name: refs/heads/master
  Altered Commit: swh:1:rev:a1b2c3d4e5f6789...
  File Path: assets/private/id_rsa
  Status: Removed
[...]
```

Solal Rapaport, Laurent Pautet, Samuel Tardieu, Stefano Zacchiroli
Altered Histories in Version Control System Repositories: Evidence from the Trenches
ASE 2025 https://arxiv.org/abs/2509.09294

Outline

1. Software as knowledge
2. Software Heritage
3. Research highlights: datasets for the masses
4. Research highlights: software evolution
5. Research highlights: diversity, equity, and inclusion
6. Research highlights: cybersecurity
7. Conclusion

Conclusion

- Software Heritage archives public code and its history as a huge Merkle DAG
- Analyzing it at scale (35/500 B nodes/edges) is a significant big-data undertaking
- Gold mine of research leads in and around empirical software engineering

Learn more

- Research: www.softwareheritage.org/publications
- Development: www.softwareheritage.org/community/developers

Student opportunities

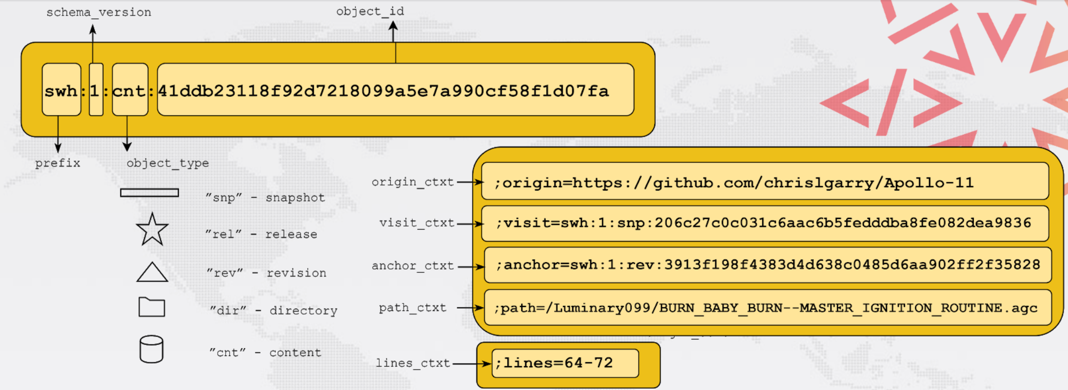
- www.softwareheritage.org/community/students

Contact

Stefano Zacchiroli / stefano.zacchiroli@telecom-paris.fr / @zacchiro@mastodon.xyz

Appendix

Software Heritage Identifiers (SWHIDs) (swhid.org)



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

An emerging standard

- Adoption: SPDX 2.2, IANA-registered "swh:" URI prefix, WikiData P6138, ...
- Breaking news: standard ISO/IEC 18670:2025

Examples

- Apollo 11 AGC excerpt
- Quake III rsqrt

An international, non profit initiative built for the long term

Sharing the vision

unesco

Software · GitLab · ENGINEERING · eclipse

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

THE LINUX FOUNDATION

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

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

FREE SOFTWARE FOUNDATION

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

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Inria

Diamond sponsors: IBM · Microsoft · TII Technology Innovation Institute

Platinum sponsors: cnrs · Ministère de l'Enseignement supérieur et de la Recherche · Infosys · HUAWEI · cea

Gold sponsors: aws · Google · openinventionnetwork · Université Paris Cité

Silver sponsors: AdaCore · Premier Ministre, Direction interministérielle du numérique · GitHub · Canadian Research Knowledge Network / Réseau canadien de documentation pour la recherche · Sorbonne Université · Università di Pisa

Bronze sponsors: Konsortium der Schweizer Hochschulbibliotheken / Consortium des bibliothèques universitaires suisses / Consorzio delle biblioteche universitarie svizzere / Consortium of Swiss Academic Libraries · NWO · Sant'Anna Scuola Universitaria Superiore Pisa · Scuola Normale Superiore · Université de Lorraine · université Paris-Saclay

www.softwareheritage.org/support/sponsors