

What Public Code Can Teach Us

Research Software meets Software Research

Stefano Zacchioli

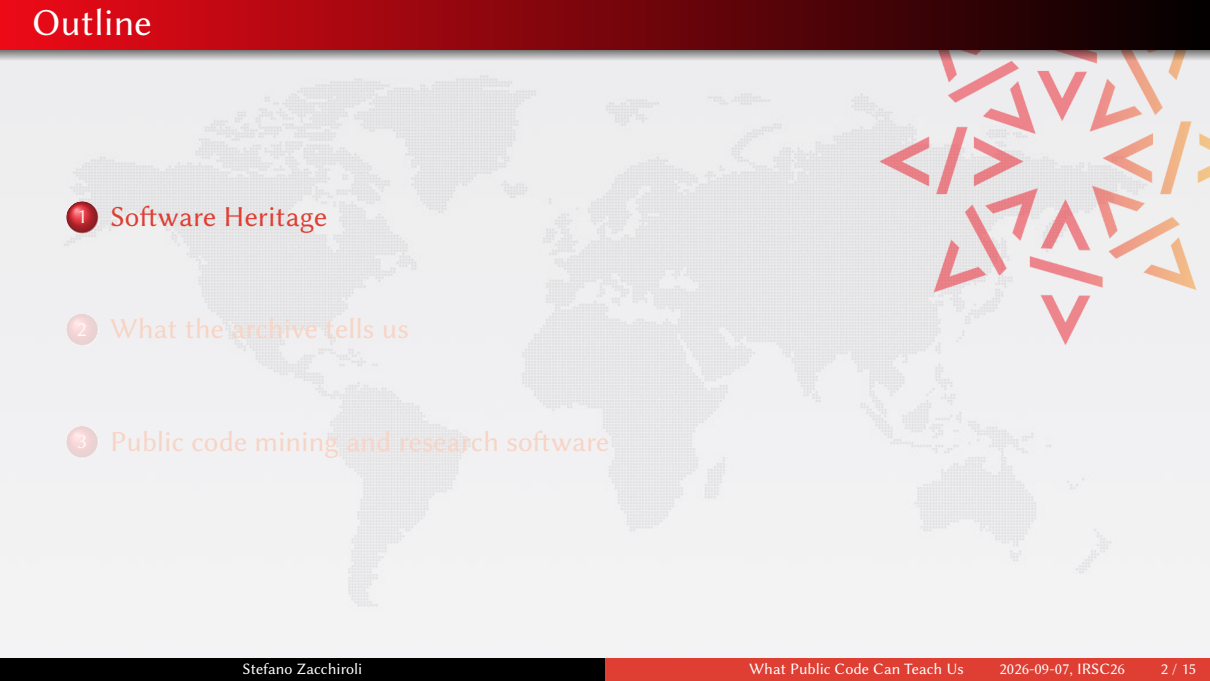
Télécom Paris, Institut Polytechnique de Paris
`stefano.zacchioli@telecom-paris.fr`

7 September 2026 — IRSC 2026, Sheffield



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

- 
- 1 Software Heritage
 - 2 What the archive tells us
 - 3 Public code mining and research software

About the speaker

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

About the speaker

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

Today's agenda

- Very brief introduction/recap of **Software Heritage**
- Showcases of **recent software research results** that leverage public code
- **Reflections** on how public code archival & analysis can help research software

Disclaimer

My expertise in research software: regular user & occasional contributor



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



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



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



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



Inria  **unesco**

The largest software archive, a shared infrastructure

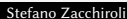
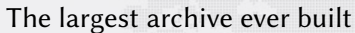
One infrastructure
open and shared



The largest archive ever built



One infrastructure
open and shared



Digital preservation of software source code, with benefits

For the first time, *all* public software is a single object of study

- **Uniform**: one data model (a Merkle DAG), forge and VCS details abstracted away: git, svn, hg, tarballs, packages all look alike
- **Deduplicated**: every file, directory, commit stored once, across 400 M+ projects: so *"where did this come from?"* becomes answerable
- **Cross-platform**: not only GitHub — also GitLab instances, university forges, Linux distros, package managers, rescued legacy code
- **Open**: the archive, the code, and all curated/derived datasets are all available (see datasets.softwareheritage.org)

Digital preservation of software source code, with benefits

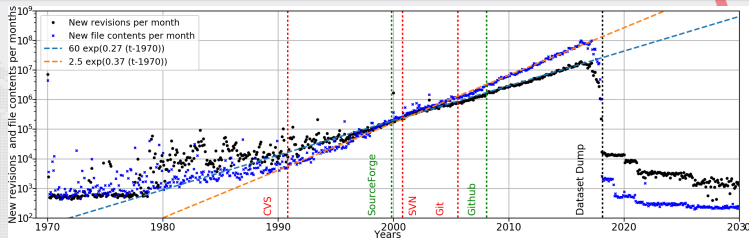
For the first time, *all* public software is a single object of study

- **Uniform**: one data model (a Merkle DAG), forge and VCS details abstracted away: git, svn, hg, tarballs, packages all look alike
- **Deduplicated**: every file, directory, commit stored once, across 400 M+ projects: so *"where did this come from?"* becomes answerable
- **Cross-platform**: not only GitHub — also GitLab instances, university forges, Linux distros, package managers, rescued legacy code
- **Open**: the archive, the code, and all curated/derived datasets are all available (see datasets.softwareheritage.org)

You are part of this

Research software included: it stops being scattered and starts being *measurable*, intermixed with the entire Software Commons.

- 
- 1 Software Heritage
 - 2 What the archive tells us
 - 3 Public code mining and research software



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, Zacchioli

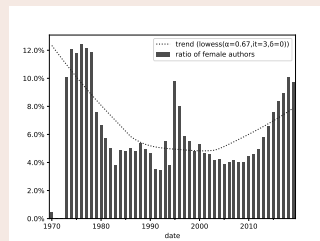
Software Provenance Tracking at the Scale of Public Source Code

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

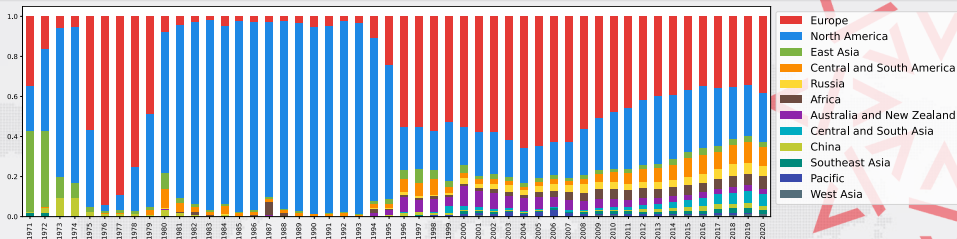
Archived commit metadata contains public information that can be mined to study diversity, equity, inclusion (DEI) traits of the global population of public code authors.

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



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.

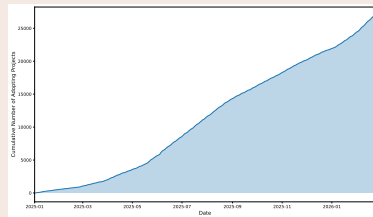
Rossi and Zacchiroli. *Geographic diversity in public code contributions*. MSR 2022

Coding agents went from nothing to a quarter of GitHub in a year

128k projects mined for traces left by coding agents:

CLAUDE.md, Co-authored-by:, copilot/* branches

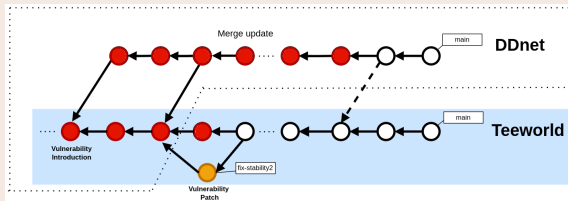
- 22–29% of projects in Jan 2025 to Feb 2026
- Broad: every topic, every language, young and old, big and small
- AI commits are $3\times$ larger; $\frac{2}{3}$ are features and bug fixes
- Follow-up replication study: projects *born after August 2025*: 72–76%, Claude Code leading



- Robbes, Matricon, Degueule, Hora, Zacchiroli. *Agentic Much? Adoption of Coding Agents on GitHub*. ACM TOSEM, 2026. Preprint: [arXiv:2601.18341](https://arxiv.org/abs/2601.18341)
- Robbes, Matricon, Degueule, Hora, Zacchiroli. *Agentic Very Much!*. Preprint: [arXiv:2606.07448](https://arxiv.org/abs/2606.07448), 2026
- Robbes, Matricon, Degueule, Hora, Zacchiroli. *Promises, Perils, and (Timely) Heuristics for Mining Coding Agent Activity*. MSR 2026

1.7 M forks left behind when a security fix lands

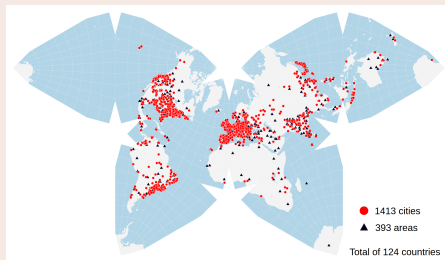
Fork between the *introduction* and the *fix* of a vulnerability, and you inherit it — silently.



- "Color" the whole graph with [OSV.dev](#) data: 7162 repos become **1.7 M vulnerable forks**
- **86.6 M vulnerable commits** specific to forks, *invisible to dependency-based tooling*
- 152 cases checked by hand: **135 high-severity vulnerabilities**, 9 confirmed by maintainers
- Only doable with a global, deduplicated data model: all forks, all platforms, one graph

- Lefeuvre, Reux, Zacchiroli, Barais, Combemale. *Chasing One-day Vulnerabilities Across Open Source Forks*. SCORED 2026. Preprint: [arXiv:2511.05097](#)

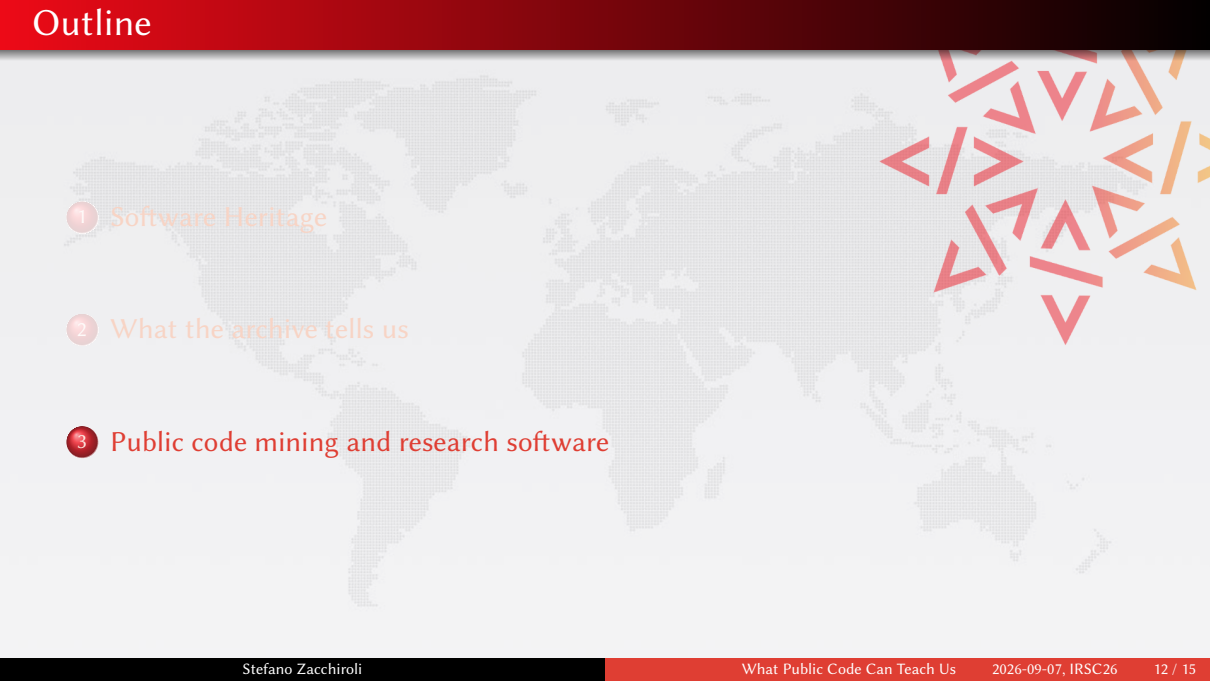
1.61 M art-related repositories mined from the archive



Found by scanning every archived origin for filesystem traces of 7 creative coding environments (Processing, p5.js, openFrameworks, SuperCollider, ...)

- 84,175 repositories (5.5%) are already offline, but preserved in Software Heritage
- Contributors in 124 countries, on every continent
- 96% of p5.js sketches synthesize new material, 50% use randomness
- Three purposes: CS education, art education, and art itself

Baudry, Gamage, Gonzalez Fernandez, MK, Pena Mendieta, Souza Pinter, Zacchiroli. *Art in Humanity's Code*. Under review, 2026. Preprint: [arXiv.org:2608.05174](https://arxiv.org/abs/2608.05174)

- 
- 1 Software Heritage
 - 2 What the archive tells us
 - 3 Public code mining and research software

Researching research software, at scale

- Research software is (often) open source → **part of the Software Commons**
- Previous findings hold for research software too, but do not **answer research-software-specific questions**.
 - E.g.: 1) What's the global evolution of research software code? 2) Who writes research software? 3) How much of research software depend on AI-assistance? 4) Are upstream security fixes timely integrated into research forks? Etc.

Researching research software, at scale

- Research software is (often) open source → **part of the Software Commons**
- Previous findings hold for research software too, but do not **answer research-software-specific questions**.
 - E.g.: 1) What's the global evolution of research software code? 2) Who writes research software? 3) How much of research software depend on AI-assistance? 4) Are upstream security fixes timely integrated into research forks? Etc.

For software researchers

- Common methodological pattern in previous studies: 1) have a selection criteria for artifacts of interest, 2) execute it on Software Heritage, 3) data analysis
 - (1) is most important: **you cannot study what you cannot identify**
 - For (2) the **heavy lifting is done already**: no scraping, no re-downloading the world (see datasets.softwareheritage.org)

Researching research software, at scale

- Research software is (often) open source → **part of the Software Commons**
- Previous findings hold for research software too, but do not **answer research-software-specific questions**.
 - E.g.: 1) What's the global evolution of research software code? 2) Who writes research software? 3) How much of research software depend on AI-assistance? 4) Are upstream security fixes timely integrated into research forks? Etc.

For software researchers

- Common methodological pattern in previous studies: 1) have a selection criteria for artifacts of interest, 2) execute it on Software Heritage, 3) data analysis
 - (1) is most important: **you cannot study what you cannot identify**
 - For (2) the **heavy lifting is done already**: no scraping, no re-downloading the world (see datasets.softwareheritage.org)

For policy makers

Support and increase efforts in the **standardization and adoption of machine-readable mechanisms** that allow to identify research software at scale (e.g., CodeMeta)

Be good citizens of scientific reproducibility

For research software practitioners (authors, researchers, engineers)

- ❶ **Archive by default.** An unarchived project cannot be studied, referenced precisely, or shown to have had impact save.softwareheritage.org
- ❷ **Reference intrinsically** with SWHIDs (Software Hash IDentifiers, ISO/IEC 18670:2025), not URLs — URLs decay, and the evidence base decays with them swhid.org
 - In particular: **use SWHIDs in your reproducibility packages**

Managing research software assets

For Open Source Program Officers (OSPOs)



ALFRED P. SLOAN
FOUNDATION



OSPO Radar (2025–): an *institutional portal* to monitor and showcase your institution's software production, *platform-agnostically*

- **curated metadata** under institutional authority; SWHIDs for persistent reference and citation
- France has HAL today; OSPO Radar brings this to *everyone else*

Managing research software assets

For Open Source Program Officers (OSPOs)



ALFRED P. SLOAN
FOUNDATION



OSPO Radar (2025–): an *institutional portal* to monitor and showcase your institution's software production, *platform-agnostically*

- **curated metadata** under institutional authority; SWHIDs for persistent reference and citation
- France has HAL today; OSPO Radar brings this to *everyone else*

A large case study — *State of Public Code: How Governments Contribute to Open Source*

With UNESCO & UN ODET: analyze every archived repo touched by a **government** email domain — 288 K repos, 193 UN member states zenodo.org/records/21258432

- The method is **one filter**: a domain registry resolved against the archive — swap in *your institution's domain* and it is **your** OSPO report
- And it surfaces what an OSPO can act on: **75% of projects carry no licence**, ~94% on a single platform, citation metadata *nearly absent*

Conclusion

- Software Heritage preserves *all* public source code **and its full history** — uniform, deduplicated, 400 M+ projects
- Preserving it made it **observable**: growth, *who* writes public code, *how* it is written now, how vulnerabilities travel across forks, what people create with code
- **Research software** is *inside* that picture, but we need to go further in **identifying it at scale**

Conclusion

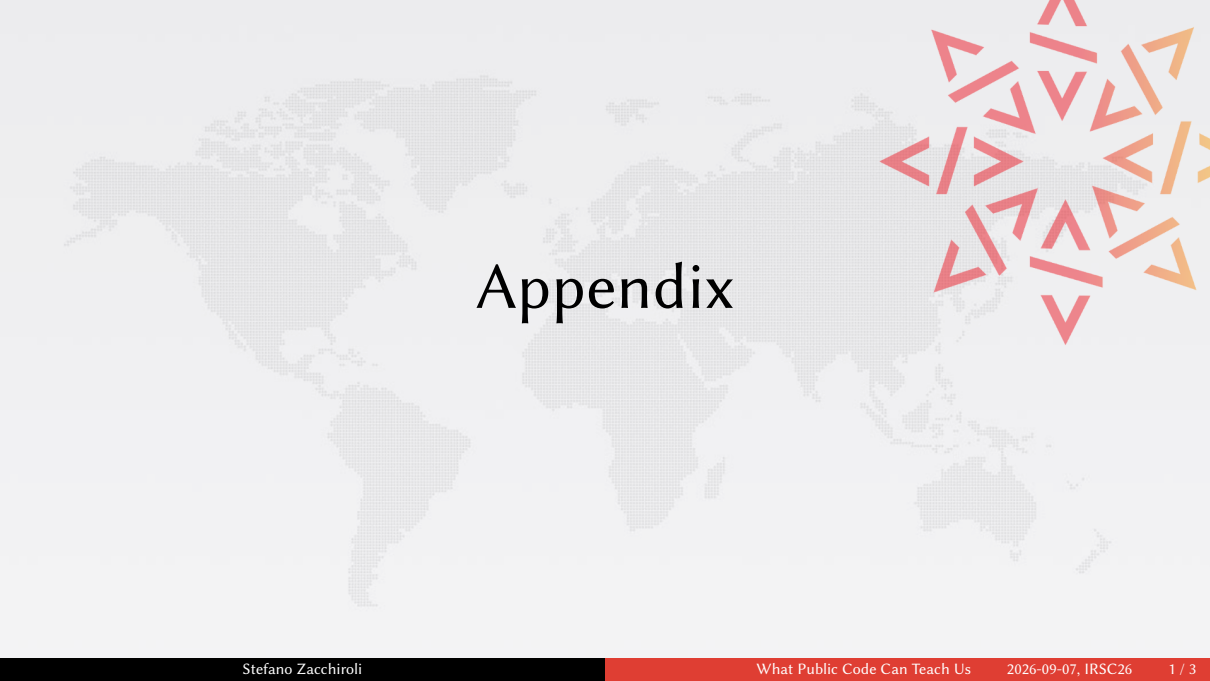
- Software Heritage preserves *all* public source code **and its full history** — uniform, deduplicated, 400 M+ projects
- Preserving it made it **observable**: growth, *who* writes public code, *how* it is written now, how vulnerabilities travel across forks, what people create with code
- **Research software** is *inside* that picture, but we need to go further in **identifying it at scale**

Fund the observatory, not just the software

Claims about the impact, traits and sustainability of research software need a *shared infrastructure* behind them. softwareheritage.org/support/sponsors

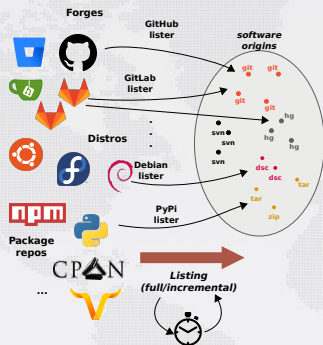
Learn more, and get in touch

- Papers: softwareheritage.org/publications — Data: datasets.softwareheritage.org
- [Stefano Zacchioli](#) / stefano.zacchioli@telecom-paris.fr / [@zacchiro@mastodon.xyz](https://mastodon.xyz/@zacchiro)

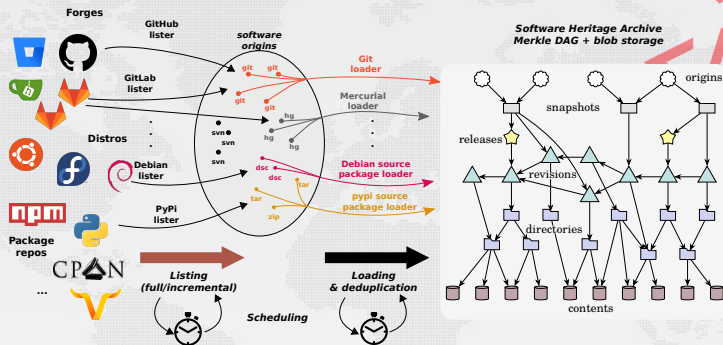


Appendix

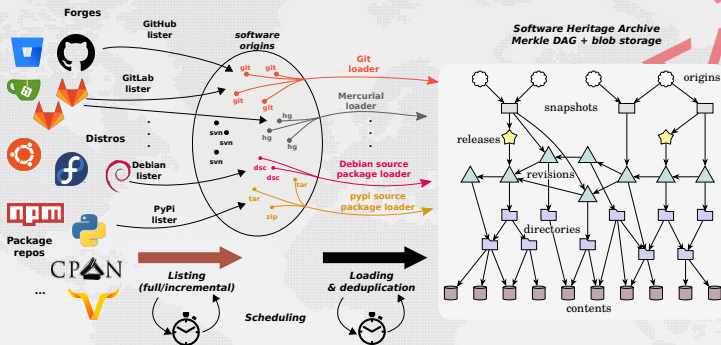
A peek under the hood: a universal archive



A peek under the hood: a universal archive

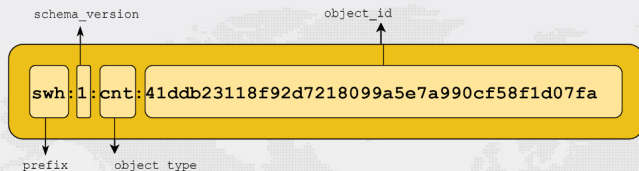


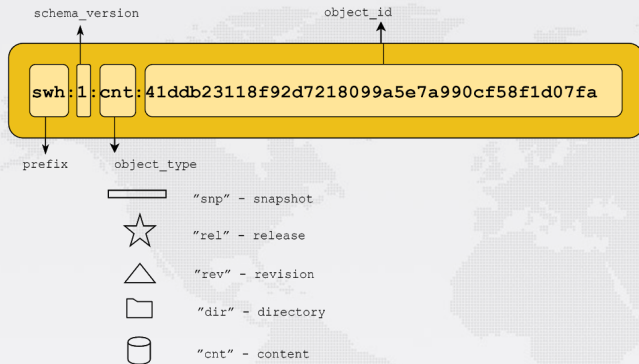
A peek under the hood: a universal archive

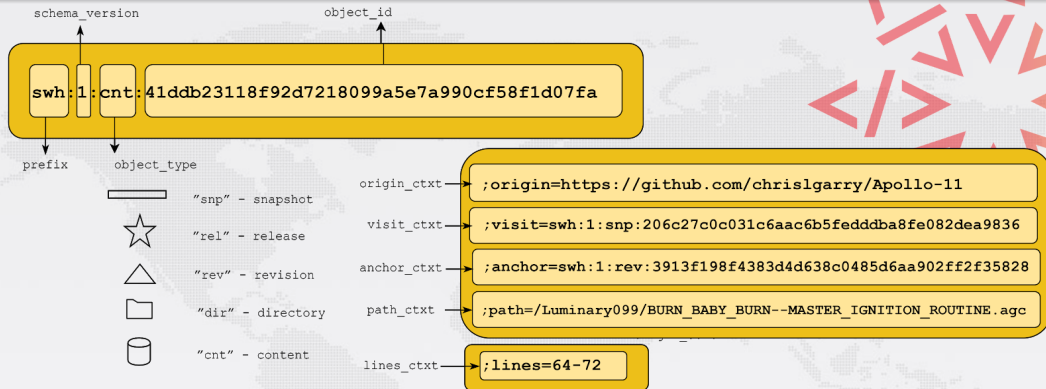


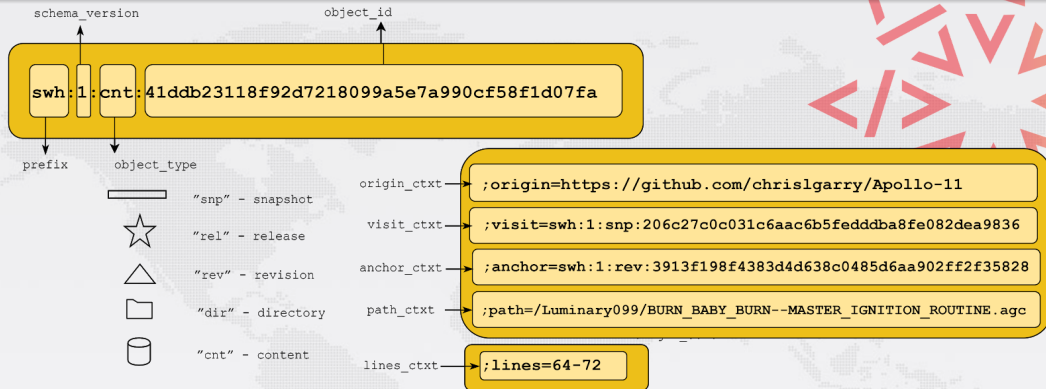
Global development history permanently archived in a uniform data model

- over 29 billion unique source files from over 438 million software projects
- ~3PB (compressed) blobs, ~60 B nodes, ~1 trillion edges



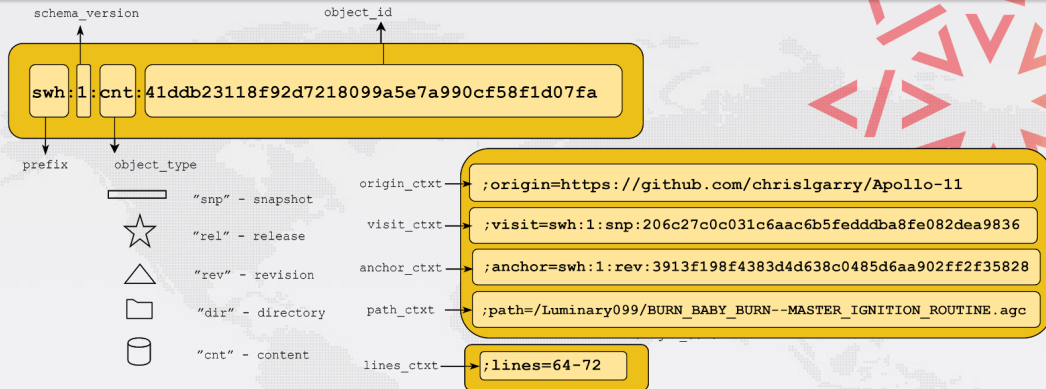






An emerging standard

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



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](#)