Software Heritage

The Great Library of Source Code

Stefano Zacchiroli

Université de Paris & Inria
zack@upsilon.cc / @zacchiro / @zacchiro@mastodon.xyz

20 March 2021
LibrePlanet

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

Our Software Commons

Definition (Commons)

The commons is the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable earth. These resources are held in common, not owned privately. https://en.wikipedia.org/wiki/Commons

Definition (Software Commons)

The software commons consists of all computer software which is available at little or no cost and which can be altered and reused with few restrictions. Thus *all open source software and all free software are part of the [software] commons.* [...]

https://en.wikipedia.org/wiki/Software_Commons

Our Software Commons

Definition (Commons)

The commons is the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable earth. These resources are held in common, not owned privately. `https://en.wikipedia.org/wiki/Commons`

Definition (Software Commons)

The software commons consists of all computer software which is available at little or no cost and which can be altered and reused with few restrictions. Thus *all open source software and all free software are part of the [software] commons.* [...]

`https://en.wikipedia.org/wiki/Software_Commons`

Source code is *a precious part* of our commons

are we taking care of it?

Software is fragile

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code, Bitbucket)
- for obsolete code: physical media decay (data rot)

Software is fragile

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code, Bitbucket)
- for obsolete code: physical media decay (data rot)

Where is the archive…

where do we go if (a repository on) GitHub or GitLab.com goes away?

Outline

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 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 all software source code

Research infrastructure

enable analysis of all software source code

Cultural Heritage | Industry | Research | Education




Software Heritage

Technology

- FOSS and transparency
- replicas all the way down

Content

- intrinsic identifiers
- facts and provenance

Organization

- non-profit
- multi-stakeholder

Archiving goals

Targets: VCS repositories & source code releases (e.g., tarballs, packages)

We DO archive

- file content (= blobs)
- revisions (= commits), with full metadata
- releases (= tags), ditto
- where (origin) & when (visit) we found any of the above

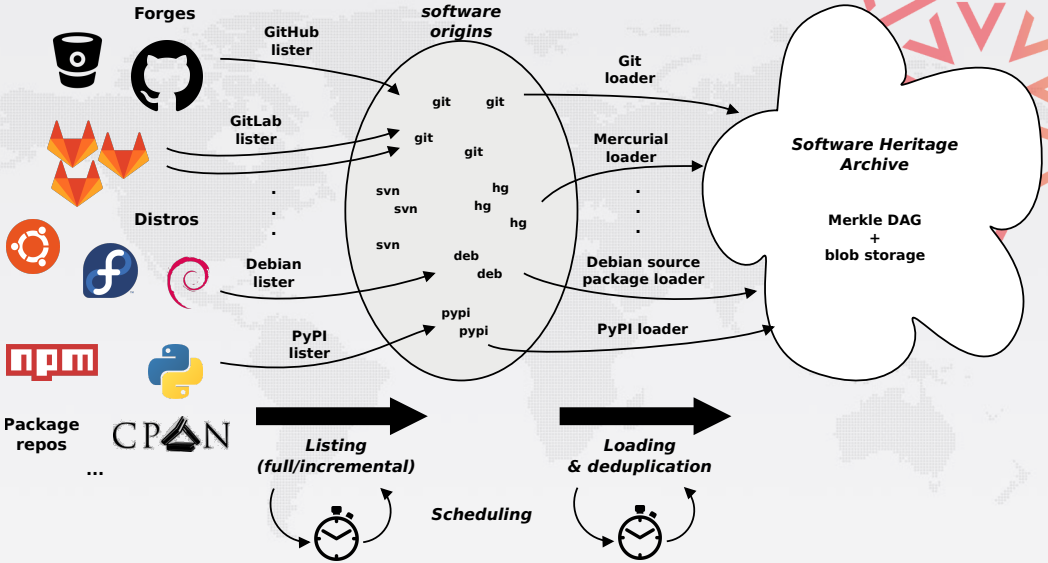
… in a VCS-/archive-agnostic canonical data model

We DON'T archive (yet)

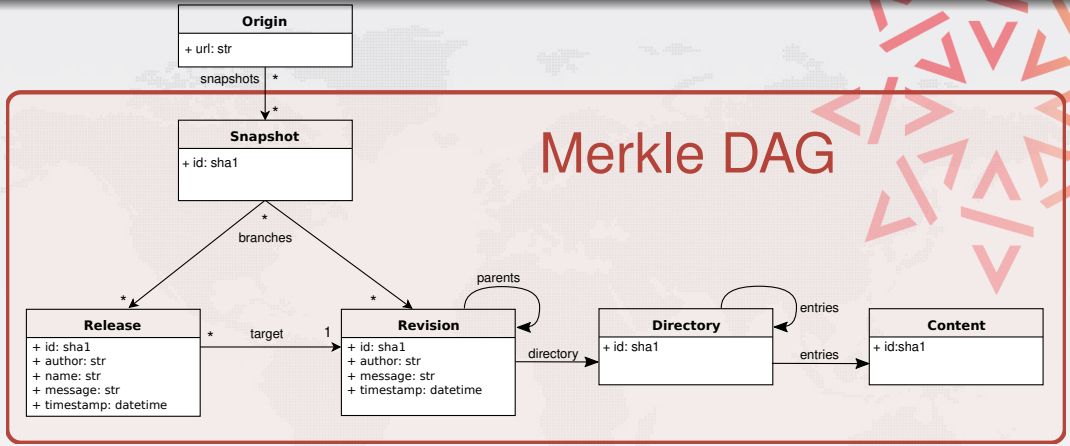
- homepages, wikis
- BTS/issues/code reviews/etc.
- mailing lists

Long term vision: play our part in a *"semantic wikipedia of software"*

Data flow

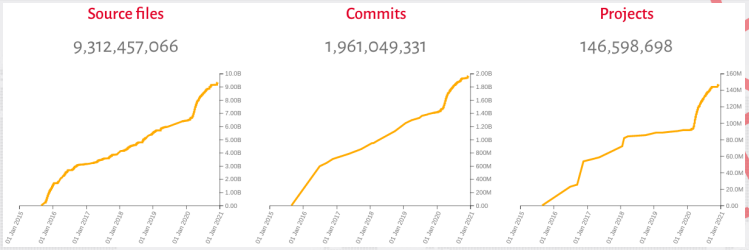

Forges
GitHub lister
GitLab lister
Distros
Debian lister
PyPI lister
Package repos
CPAN
...
software origins
git
git
git
git
svn
svn
svn
hg
hg
hg
deb
deb
pypi
pypi
Git loader
Mercurial loader
Debian source package loader
PyPI loader
Software Heritage Archive
Merkle DAG
+
blob storage
Listing (full/incremental)
Scheduling
Loading & 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*.

Archive coverage — archive.softwareheritage.org

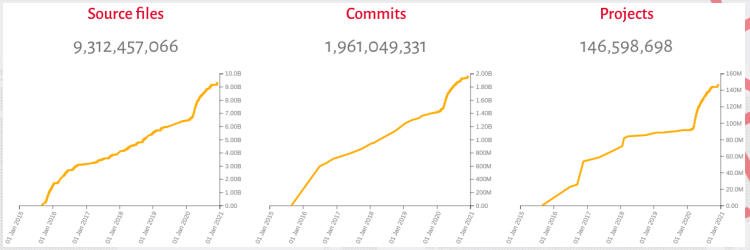


Bitbucket · R · debian · Framagit · GitHub · GitLab

GITORIOUS · Google code · GNU · Guix · HAL archives-ouvertes.fr · Inria

IPOL Journal · npm · NixOS · python

Archive coverage — archive.softwareheritage.org



Bitbucket · R · debian · Framagit · GitHub · GitLab

GITORIOUS · Google code · {P} · GNU · Guix · HAL archives-ouvertes.fr · Inria

IPOL Journal · npm · NixOS · python

- on disk: ~700 TB (uncompressed); as a graph ~20 B nodes, ~200 B edges
- the largest public source code archive in the world (and growing!)

Software Heritage and GNU Guix

Software Heritage and GNU Guix join forces to enable long term reproducibility

Connecting reproducible deployment to a long-term source code archive

Ludovic Courtès — March 29, 2019

GNU Guix can be used as a "package manager" to install and upgrade software packages as is familiar to GNU/Linux users, or as an environment manager, but it can also provision containers or virtual machines, and manage the operating system running on your machine.

One foundation that sets it apart from other tools in these areas is *reproducibility*. From a high-level view, Guix allows users to *declare* complete software environments and instantiate them. They can share those environments with others, who can replicate them or adapt them to their needs. This aspect is key to reproducible computational experiments: scientists need to reproduce software environments before they can reproduce experimental results, and this is one of the things we are focusing on in the context of the Guix-HPC effort. At a lower level, the project, along with others in the Reproducible Builds community, is working to ensure that software build outputs are reproducible, bit for bit.

Work on reproducibility at all levels has been making great progress. Guix, for instance, allows you to travel back in time. That Guix can travel back in time *and* build software reproducibly is a great step forward. But there's still an important piece that's missing to make this viable: a stable source code archive. This is where Software Heritage (SWH for short) comes in.

When source code vanishes

- https://www.softwareheritage.org/2019/04/18/software-heritage-and-gnu-guix-join-forces-to-enable-long-term-reproducibility
- https://guix.gnu.org/blog/2019/connecting-reproducible-deployment-to-a-long-term-source-code-archive/

Outline

Web interface

archive.softwareheritage.org

DEMO TIME !

Web API

RESTful API to programmatically access the Software Heritage archive
`https://archive.softwareheritage.org/api/`

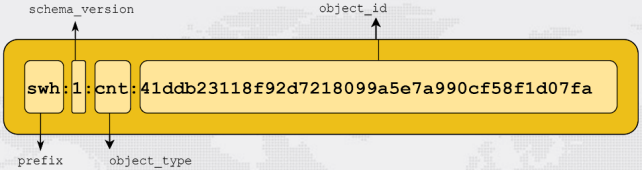
Features

- pointwise browsing of the archive
  - … snapshots → revisions → directories → contents …
- full access to the metadata of archived objects
- crawling information
  - *when have you last visited this Git repository I care about?*
  - *where were its branches/tags pointing to at the time?*

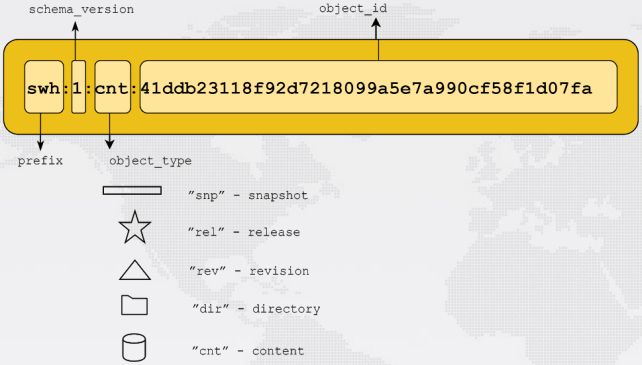
Endpoint index

`https://archive.softwareheritage.org/api/1/`

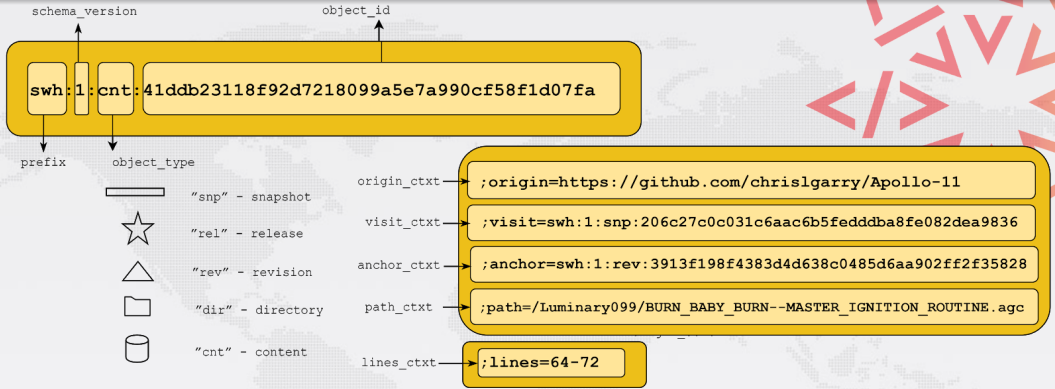
Software Heritage Identifiers (SWHIDs) (full spec)



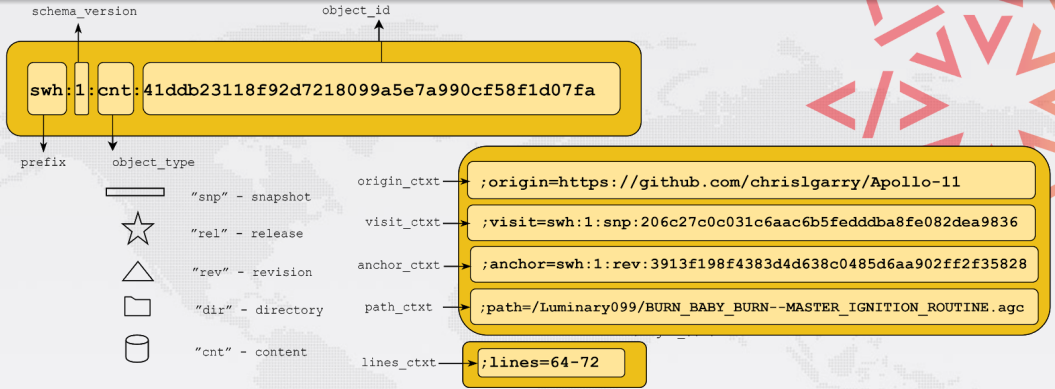
Software Heritage Identifiers (SWHIDs) (full spec)



Software Heritage Identifiers (SWHIDs) (full spec)



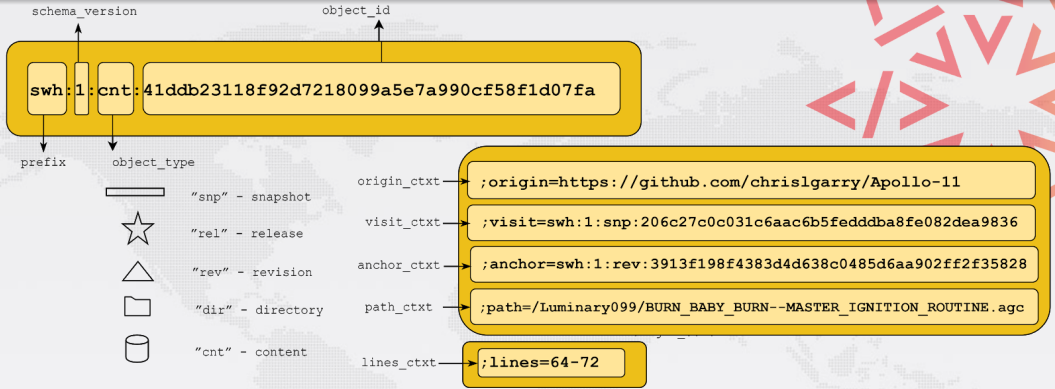
Software Heritage Identifiers (SWHIDs) (full spec)



An emerging standard

- in Linux Foundation's SPDX 2.2
- IANA-registered "swh:" URI prefix
- WikiData property P6138

Software Heritage Identifiers (SWHIDs) (full spec)



An emerging standard

- in Linux Foundation's SPDX 2.2
- IANA-registered "`swh:`" URI prefix
- WikiData property P6138

Examples

- Apollo 11 AGC excerpt
- Quake III rsqrt

```
$ pip install swh-model[cli]

$ swh identify fork.c kmod.c sched/deadline.c
swh:1:cnt:2e391c754ae730bd2d8520c2ab497c403220c6e3  fork.c
swh:1:cnt:0277d1216f80ae1adeed84a686ed34c9b2931fc2  kmod.c
swh:1:cnt:57b939c81bce5d06fa587df8915f05affbe22b82  sched/deadline.c

$ swh identify --no-filename /usr/src/linux/kernel/
swh:1:dir:f9f858a48d663b3809c9e2f336412717496202ab

$ git clone --mirror \
    https://forge.softwareheritage.org/source/helloworld.git
$ swh identify --type snapshot helloworld.git/
swh:1:snp:510aa88bdc517345d258c1fc2babcd0e1f905e93  helloworld.git
```

Warning

If you expect *others* to be able to resolve the SWHIDs of source code you care about, you should make sure the corresponding software is archived in Software Heritage.

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:
  `https://forge.softwareheritage.org/source/swh-fuse/`
- documentation:
  `https://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: The 43rd International Conference on Software Engineering
`https://arxiv.org/abs/2102.06390`

Software Heritage Filesystem (SwhFS) — Tutorial

```
$ pip install swh.fuse  # install SwhFS

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

$ ls -1F swhfs/  # list entry points
archive/  # <- start browsing from here
cache/
origin/
README
```

Software Heritage Filesystem (SwhFS) — Tutorial (cont.)

```
$ cd swhfs/

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

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

Software Heritage Filesystem (SwhFS) — Tutorial (cont.)

```
$ 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
```

Software Heritage Filesystem (SwhFS) — Tutorial (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
```

Software Heritage Filesystem (SwhFS) — Tutorial (cont.)

```
$ swh web search git-annex --limit 1
...
git://git.joeyh.name/git-annex.git \
  https://archive.softwareheritage.org/api/1/origin/git://git.joeyh.name/g
...

$ swh web search git-annex --url-encode | cut -f 1
git%3A%2F%2Fgit.joeyh.name%2Fgit-annex.git

$ cd origin/git%3A%2F%2Fgit.joeyh.name%2Fgit-annex.git
$ ls -F
2020-12-19/

$ ls 2020-12-19/snapshot/refs/heads/master/root/
Annex/                  COPYRIGHT           NEWS
Annex.hs                Creds.hs            P2P/
Assistant/              Crypto.hs           README
Assistant.hs            Database/           Remote/
Backend/                debian/             RemoteDaemon/
```

Outline

You can help!

Financially

- Donations: www.softwareheritage.org/donate/
- Sponsoring: www.softwareheritage.org/support/sponsors/

Coding

- Developer info: www.softwareheritage.org/community/developers/

Expanding archive coverage

- save.softwareheritage.org
- Development grants: framaforms.org/expression-of-interest-expanding-software-heritage-1589195728

Student opportunities

- Internships: wiki.softwareheritage.org/wiki/Internships
- Google Summer of Code 2021: wiki.softwareheritage.org/wiki/gsoc

Q: do you archive *only* Free Software?

- We only crawl origins *meant* to host source code (e.g., forges)
- Most (~90%) of what we *actually* retrieve is textual content

Our goal

Archive the entire Free Software Commons, present and future.

- Large parts of what we retrieve is *already* Free Software, today
- Most of the rest *will become* Free Software in the long term
  - e.g., at copyright expiration