Towards an Open Data and Open Source Code Scanner

for your Open Compliance

Stefano Zacchiroli

Software Heritage — zack@upsilon.cc, @zacchiro

7 April 2021
Legal & Licensing Workshop
online

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

About the speaker

- Associate Professor of Computer Science, Université de Paris, on leave at Inria
- Free/Open Source Software activist (20+ years)
- Debian Developer & Former 3x Debian Project Leader
- Former Open Source Initiative (OSI) director
- Software Heritage co-founder & CTO

Outline

Open Compliance

My own take at a definition of a notion many of us care about:

Definition (Open Compliance)

The pursuit of compliance with *license obligations* and other *best practices* for the management of open source software components, using only open technologies such as: open source software, open data information, and open access documentation.

Open Compliance

My own take at a definition of a notion many of us care about:

Definition (Open Compliance)

The pursuit of compliance with *license obligations* and other *best practices* for the management of open source software components, using only open technologies such as: open source software, open data information, and open access documentation.

Why

- Reduced lock-in risks
- Lower total cost of ownership (TCO)
- Allow to crowdsource expensive compliance steps (e.g., scanning, curation)
- Aligned with the ethos of free/open source software (FOSS) communities

Long-discussed in FOSS compliance circles. Many well-established collaboration initiatives: Open Source Tooling Group, Open Compliance Program, Double Open, …

Reuse is the new rule

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

Where does reused software come from?

Reuse is the new rule

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

Where does reused software come from?

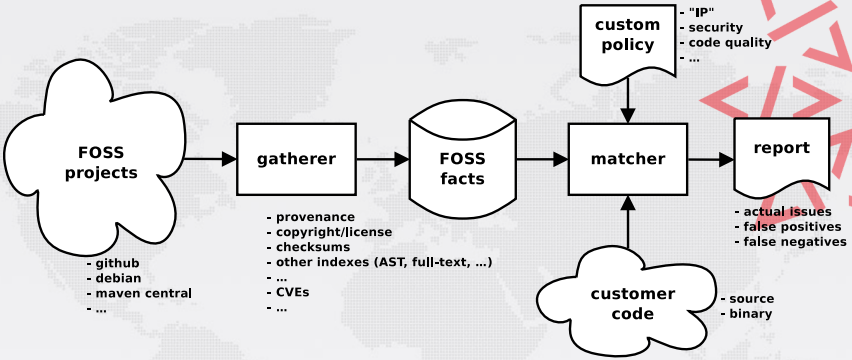
Do *you* know where it comes from?

- the software you ship
- the software you use
- the software you acquire
- the software that
  - has that bug
  - has that vulnerability

KYSW: Know Your SoftWare

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

Anatomy of a KYSW toolchain



source: *A Community Take on the License Compliance Industry*, Stefano Zacchiroli, FOSDEM 2016, Legal and Policy Issues devroom,

`https://upsilon.cc/~zack/talks/2016/2016-01-31-fosdem-compliance.pdf`

A source code scanner is the key ingredient of all KYSW toolchains: it scans a local *source* code base and compares it to a FOSS knowledge base, summarizing findings.

(We will ignore other features for the purpose of this talk.)

An Open Compliance Source Code Scanner — Requirements

Be Open Compliance-…compliant

- front-end: open source client, running locally on your code base
- back-end: open data knowledge base, either remote or self-hosted

Practical needs

- known/unknown information (has this been published before?)
- license information
- provenance information
- scanning granularity: both file-level and snippet-level
- knowledge-base coverage: cover all of FOSS

An Open Compliance Source Code Scanner — Requirements

Be Open Compliance-…compliant

- front-end: open source client, running locally on your code base
- back-end: open data knowledge base, either remote or self-hosted

Practical needs

- known/unknown information (has this been published before?)
- license information
- provenance information
- scanning granularity: both file-level and snippet-level
- knowledge-base coverage: cover all of FOSS

Claim: we still lack a source code scanning tool that is compliant with Open Compliance principles and addresses industry practical needs.

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

A principled infrastructure

http://bit.ly/swhpaper

A principled infrastructure

http://bit.ly/swhpaper

Technology

- transparency and FOSS
- replicas all the way down

Content

- intrinsic identifiers
- facts and provenance

Organization

- non-profit
- mirror network

Sharing the vision

And many more ...

www.softwareheritage.org/support/testimonials

An international, non profit initiative built for the long term

Sharing the vision

United Nations Educational, Scientific and Cultural Organization

And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Inria — INVENTEURS DU MONDE NUMÉRIQUE

Platinum sponsors

cnrs · intel · Microsoft · Ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation · HUAWEI · SOCIETE GENERALE

Gold sponsors

openinventionnetwork · Université de Paris · SORBONNE UNIVERSITÉ

Silver sponsors

CASE · Liberté • Égalité • Fraternité RÉPUBLIQUE FRANÇAISE · GitHub · Google · Università di Pisa · vmware

Bronze sponsors

DANS · FOSSID

The largest free/open source software archive

Bitbucket · R · debian · FramaGit · GitHub · GitLab

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

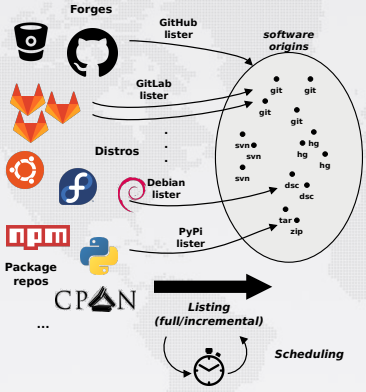
IPOL Journal · npm · NixOS · python Software Foundation

The largest free/open source software archive

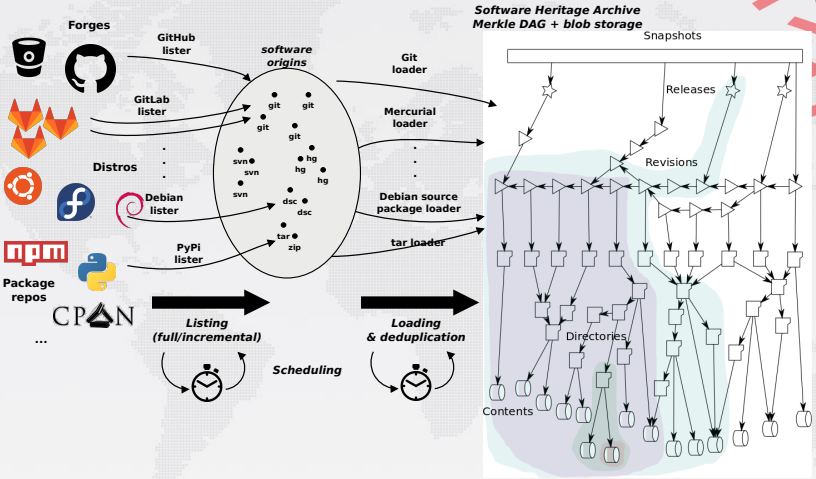
Bitbucket · R · debian · FramaGit · GitHub · GitLab · GITORIOUS · Google code · {P} · GNU · Guix · HAL archives-ouvertes.fr · Inria · IPOL Journal · npm · NixOS · python

- 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!)

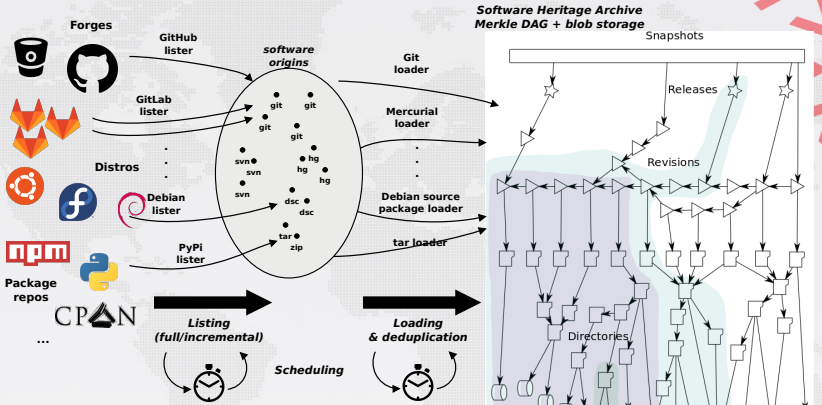
Automation and storage



Automation and storage

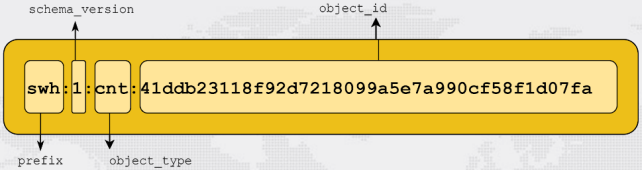


Automation and storage

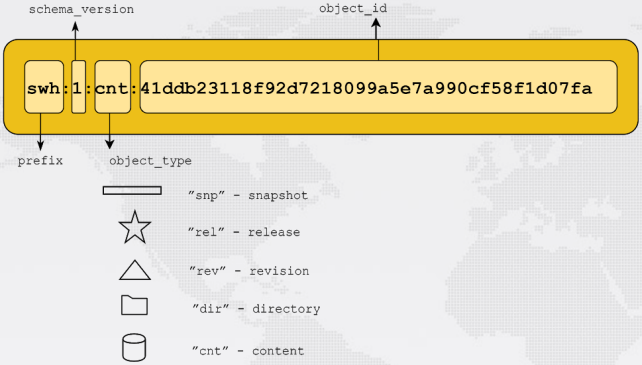


Full development history permanently archived in a uniform data model.

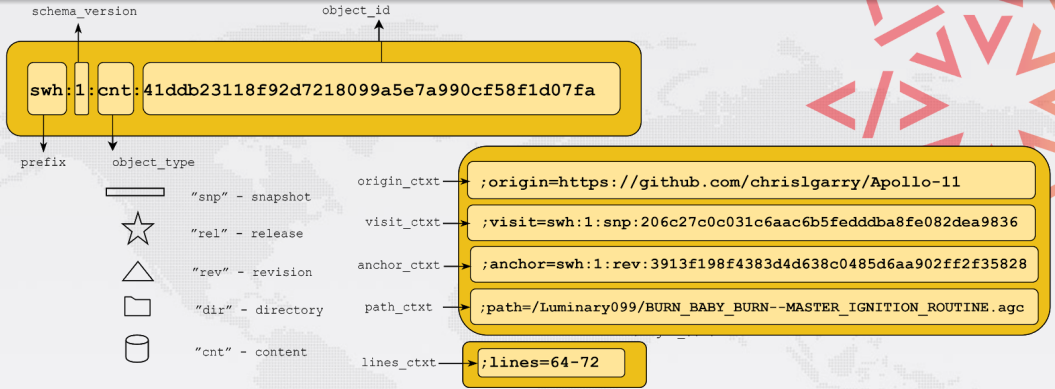
Meet the Software Heritage Identifiers (SWHIDs) (full spec)



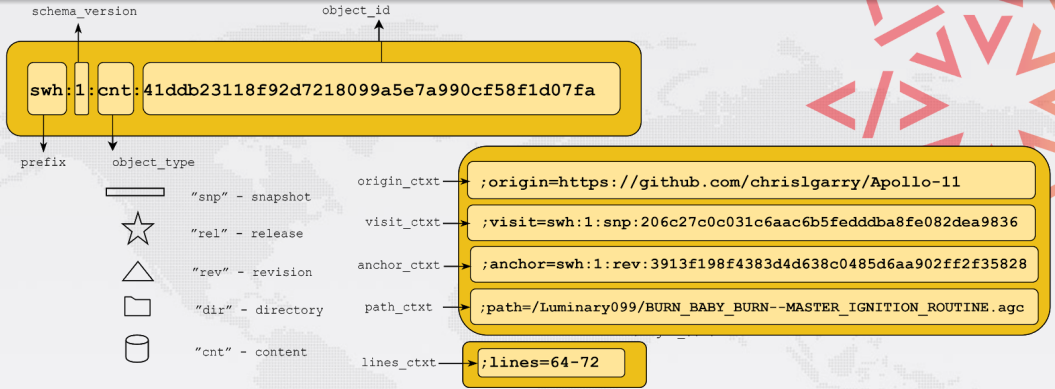
Meet the Software Heritage Identifiers (SWHIDs) (full spec)



Meet the Software Heritage Identifiers (SWHIDs) (full spec)



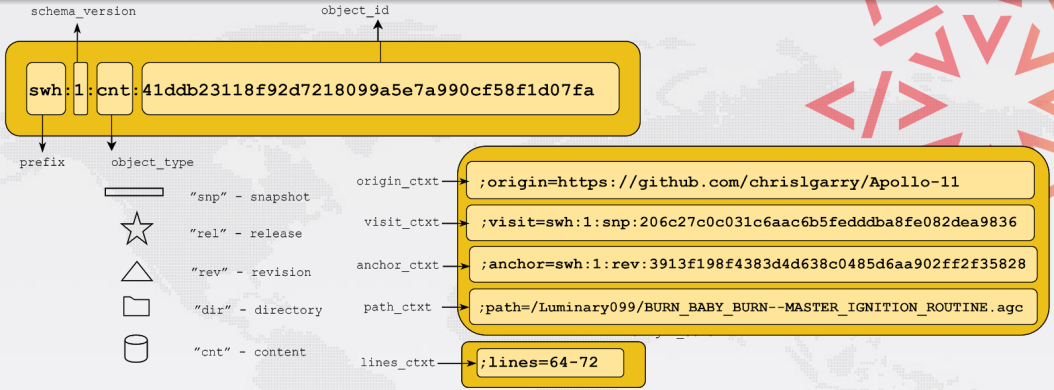
Meet the Software Heritage Identifiers (SWHIDs) (full spec)



An emerging standard

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

Meet the Software Heritage Identifiers (SWHIDs) (full spec)



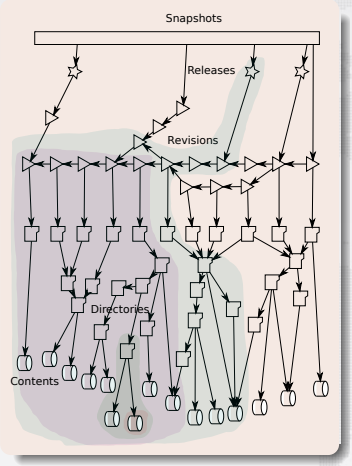
An emerging standard

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

Examples

- Apollo 11 AGC excerpt
- Quake III rsqrt

"It's ~~Turtles~~ SWHIDs all the way down"



Reference *any* source code artifact that has ever been shared—source code file, tree, commit, release, repository state—using the same, standard identifier.

Try it out:

```
$ pip install swh.model[cli]
$ swh identify /srv/src/linux/kernel/
swh:1:dir:b770a2aed8db52df737f88f18ca6bf39a1582240
```

Outline

Tech preview: swh-scanner

Vision

`swh-scanner` is an open source and open data source code scanner for open compliance workflows, backed by the largest public archive of FOSS source code.

Tech preview: swh-scanner

Vision

`swh-scanner` is an open source and open data source code scanner for open compliance workflows, backed by the largest public archive of FOSS source code.

Design (of the current prototype)

- query the Software Heritage archive as source of truth about public code
- leverages the Merkle DAG model and SWHIDs for maximum scanning efficiency
  - e.g., no need to query the back-end for files contained in a known directory
- file-level granularity
- output: source tree partition into known (= published before) v. unknown

code: forge.softwareheritage.org/source/swh-scanner (GPL 3+)
package: pypi.org/project/swh.scanner

swh-scanner — Demo

```
$ pip install swh.scanner

$ swh scanner scan -f json /srv/src/linux/kernel
{
  [...]
  "/srv/src/linux/kernel/auditsc.c": {
      "known": true,
      "swhid": "swh:1:cnt:814406a35db163080bbf937524d63690861ff750" },
  "/srv/src/linux/kernel/backtracetest.c": {
      "known": true,
      "swhid": "swh:1:cnt:a2a97fa3071b1c7ee6595d61a172f7ccc73ea40b" },
  "/srv/src/linux/kernel/bounds.c": {
      "known": true,
      "swhid": "swh:1:cnt:9795d75b09b2323306ad6a058a6350a87a251443" },
  "/srv/src/linux/kernel/bpf": {
      "known": true,
      "swhid": "swh:1:dir:fcd9987804d26274fee1eb6711fac38036ccaee7" },
  "/srv/src/linux/kernel/capability.c": {
      "known": true,
      "swhid": "swh:1:cnt:1444f3954d750ba685b9423e94522e0243175f90" },
  [...]
}
0,53s user 0,61s system 145% cpu 1,867 total
```

swh-scanner — Demo (cont.)

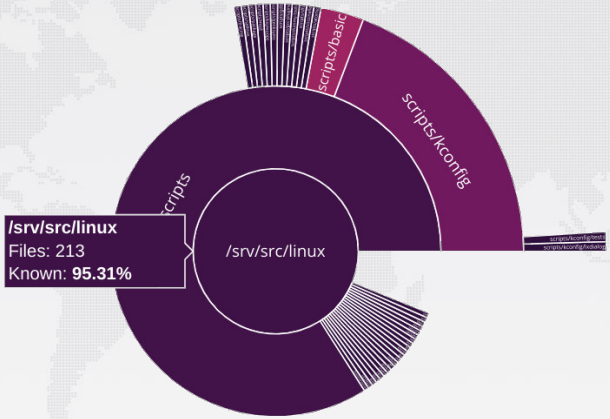
```
$ du -sh --exclude=.git /srv/src/linux
1,1G /srv/src/linux

$ time swh scanner scan -f json -x *.git /srv/src/linux
{
  [...]
  "/srv/src/linux/arch": {
      "known": true,
      "swhid": "swh:1:dir:590c329d3548b7d552fc913a51965353f01c9e2f" },
  [...]
  "/srv/src/linux/scripts/kallsyms.c": {
      "known": true,
      "swhid": "swh:1:cnt:0096cd9653327584fe62ce56ba158c68875c5067" },
  "/srv/src/linux/scripts/kconfig": {
      "known": false,
      "swhid": "swh:1:dir:548afc93bd01d2fba0dfcc0fd8c69f4b082ab8c6" },
  "/srv/src/linux/scripts/kconfig/.conf.o.cmd": {
      "known": false,
      "swhid": "swh:1:cnt:0d8be19e430c082ece6a3803923ad6ecb9e7d413" },
  [...]
}
20,84s user 1,52s system 103% cpu 21,540 total
```

swh-scanner — Demo (cont.)

Interactive mode to drill-down and inspect unknown files:

```
$ swh scanner scan -f sunburst -x *.git /srv/src/linux
```



Outline

swh-scanner — Checklist

Open Compliance

- ✓ front-end: open source client, running locally on your code base
- ✓ back-end: open data knowledge base, remote or self-hosted

swh-scanner — Checklist

Open Compliance

- ✓ front-end: open source client, running locally on your code base
- ✓ back-end: open data knowledge base, remote or self-hosted

Practical needs

- ✓ known/unknown information (has this been published before?)
- ✗ license information
- ✗ provenance information
- ✓ file-level granularity
- ✗ snippet-level granularity
- ✓ knowledge-base coverage: all of ~~FOSS~~ Software Heritage

swh-scanner — Going further

swh-scanner shows that *it is possible* to create a source code scanner that is both open source and backed by the most comprehensive open data FOSS archive.

swh-scanner — Going further

swh-scanner shows that *it is possible* to create a source code scanner that is both open source and backed by the most comprehensive open data FOSS archive.

Roadmap

swh-scanner is *not a production-ready scanner*. The following features are still missing:

- license information → in-house scanning + ClearlyDefined
- provenance information → Software Heritage crawling info
- increase granularity to snippet/SLOC

Some of these are low-hanging fruits, some require substantial R&D investments.

swh-scanner — Going further

swh-scanner shows that *it is possible* to create a source code scanner that is both open source and backed by the most comprehensive open data FOSS archive.

Roadmap

swh-scanner is *not a production-ready scanner*. The following features are still missing:

- license information → in-house scanning + ClearlyDefined
- provenance information → Software Heritage crawling info
- increase granularity to snippet/SLOC

Some of these are low-hanging fruits, some require substantial R&D investments.

Feedback welcome

- feel free to play with swh-scanner, feedback is very welcome!
- caveat: intensive use will result in hitting the API rate-limit

Getting involved

For those interested in depositing and tracking software source code.

Source code deposit interest group (DIG)

Benefits / Level	strategic	core	solutions	basic
strategic advisory	Y			
technical advisory	Y	Y		
general assembly	Y	Y	Y	
deposit code	Y	Y	Y	Y

Current members

OIN, VMware, DINUM, CNRS, MESRI, University of Paris

How to join

contact us at sponsor@softwareheritage.org

Wrapping up

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

www.softwareheritage.org @swheritage

- open compliance is about FOSS management using *only* open technology
- we still lack a fully open—open source, backed by an open data knowledge base—source code scanner for open compliance toolchains
- swh-scanner is a *prototype scanner* showing that it is possible, today, to develop such a scanner, building on Software Heritage as an extensive knowledge base
- swh-scanner is not an industry-ready scanner, but might become one; its architecture and components can be reused elsewhere

Contacts

Stefano Zacchiroli / zack@upsilon.cc / @zacchiro / @zacchiro@mastodon.xyz

Complete Corresponding Source (CCS) hosting

Complete Corresponding Source (CCS) requirement

For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. — *GPLv2*

CCS management in the real world

- CCS tarballs published at release time; URLs included in user manuals
- IT reorganizations → link rot (e.g., 404 on CCS URLs) → out of compliance

Complete Corresponding Source (CCS) hosting

Complete Corresponding Source (CCS) requirement

> *For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable.* — *GPLv2*

CCS management in the real world

- CCS tarballs published at release time; URLs included in user manuals
- IT reorganizations → link rot (e.g., 404 on CCS URLs) → out of compliance

A better approach (Intel+SWH prototype)

Delegate CCS hosting to an archive:

1. prepare CCS tarball
2. deposit it to Software Heritage
3. include SWHID in user manuals

Complete Corresponding Source (CCS) hosting

Complete Corresponding Source (CCS) requirement

> *For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable.* — *GPLv2*

CCS management in the real world

- CCS tarballs published at release time; URLs included in user manuals
- IT reorganizations → link rot (e.g., 404 on CCS URLs) → out of compliance

A better approach (Intel+SWH prototype)

Delegate CCS hosting to an archive:

1. prepare CCS tarball
2. deposit it to Software Heritage
3. include SWHID in user manuals

Is it compliant?

- TL;DR: yes! (with agreement with hoster)
- Cf. GPL FAQ *Can I put the binaries on my Internet server and put the source on a different Internet site?*

Depositing source code to Software Heritage

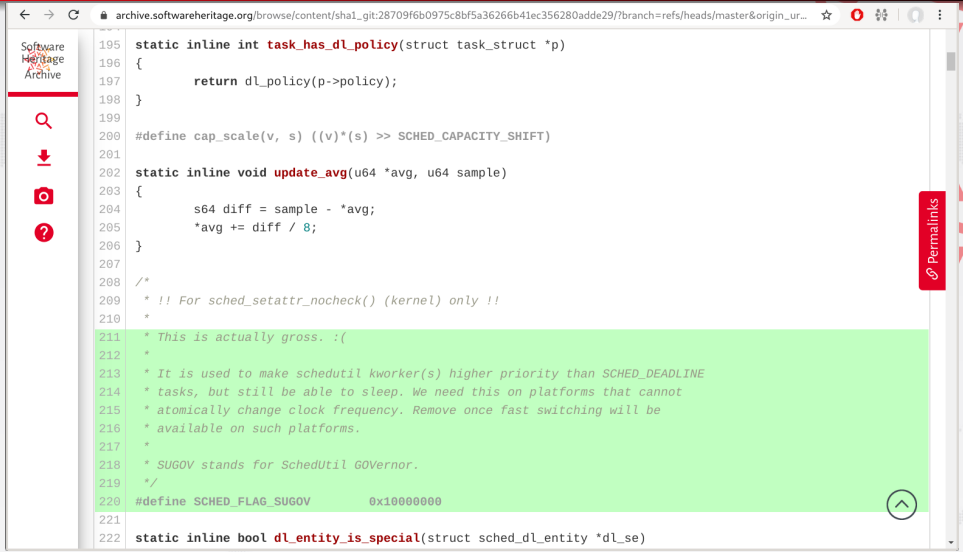
Deposit service

- complement regular (pull) crawling of forges and distributions
- restricted access (i.e., not a warez dumpster!)
- deposit.softwareheritage.org

Tech bits

- SWORD 2.0 compliant server, for digital repositories interoperability
- RESTful API for deposit and monitoring, with CLI wrapper

Web UI — Browse the Great Library of Source Code



https://archive.softwareheritage.org / <SWHID>

Web API — Integrate your tools with the Software Heritage archive

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/