Building a Safer Open Source Supply Chain on top of the Software Heritage Rock

Stefano Zacchiroli

Software Heritage
Télécom Paris, Polytechnic Institute of Paris

1 June 2023
Sécurité de la Supply Chain Logicielle
Paris, France

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline

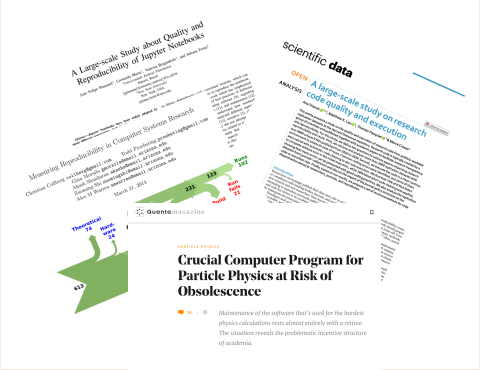
About the speaker

- Professor of Computer Science, Télécom Paris, Polytechnic Institute of 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 & CTO
- Reproducible Builds board member

Outline

How are we managing our software ?

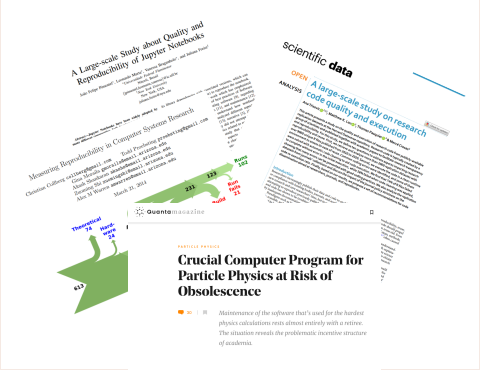
Reproducibility, maintenance in Academia



(articles: here, here, here and here)

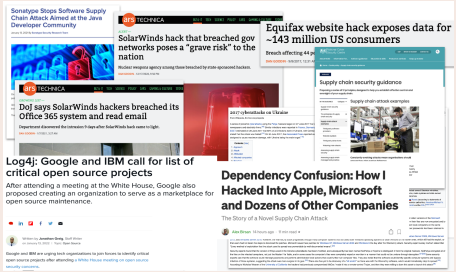
How are we managing our software ?

Reproducibility, maintenance in Academia



(articles: here, here, here and here)

Security, integrity, traceability in Industry

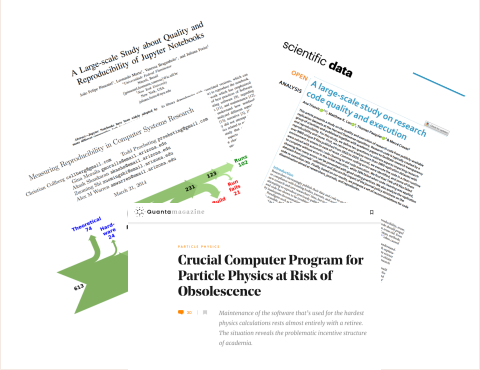


Can they track the software that they

- ship, use, acquire
- has that bug or vulnerability

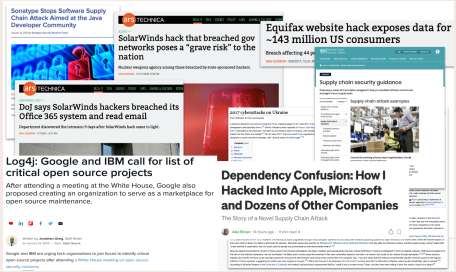
How are we managing our software ?

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

A few basic needs for software in Open Science (and Industry)

Archive

Research software artifacts must be properly archived

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

A few basic needs for software in Open Science (and Industry)

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*)

A few basic needs for software in Open Science (and Industry)

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*)

A few basic needs for software in Open Science (and Industry)

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*!)

A few basic needs for software in Open Science (and Industry)

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*!)

These are also industry needs!

Open Source is growing…

Software is eating the world

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 traditional companies

Marc Andreesen, 2011

Open Source is growing…

Software is eating the world

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 traditional companies

Marc Andreesen, 2011

Open Source is eating the Software World

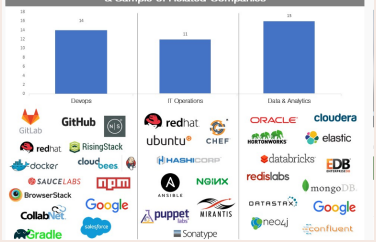
TC News Startups Mobile Gadgets Enterprise Social Europe Trend

CRUNCH NETWORK

Tracking the explosive growth of open-source software

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

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



Open Source is growing…

Software is eating the world

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

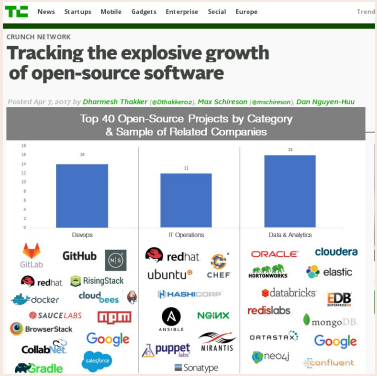
August 20, 2011

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 traditional companies

Marc Andreesen, 2011

Open Source is eating the Software World



TC News Startups Mobile Gadgets Enterprise Social Europe Trend

CRUNCH NETWORK

Tracking the explosive growth of open-source software

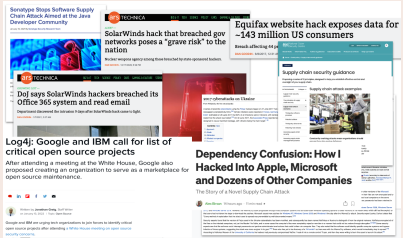
Posted Apr 7, 2017 by Dharmesh Thakker (@dthakkerca), Max Schireson (@mschireson), Dan Nguyen-Huu

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

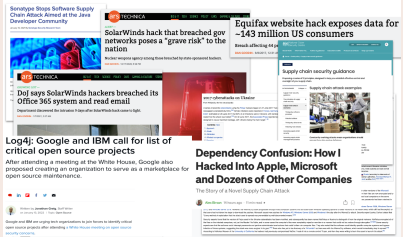
Reuse is the new rule

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

Software supply chain attacks abound



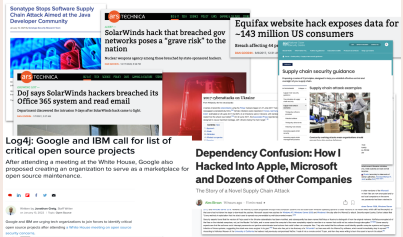
Software supply chain attacks abound



Can you track the software that…

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

Software supply chain attacks abound



Can you track the software that…

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

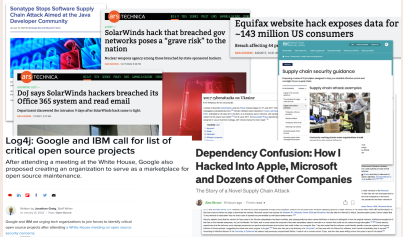
KYSW: Know Your SoftWare - like KYC in banking

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

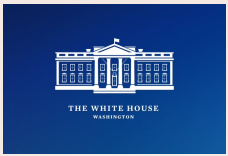
Software supply chain attacks abound



Can you track the software that…

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

KYSW: Know Your SoftWare - like KYC in banking



Sec. 4. Enhancing Software Supply Chain Security

ensuring and attesting, to the extent practicable, to the integrity and provenance of open source software

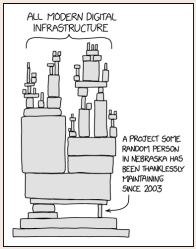
May 2021 POTUS Executive Order

Can we fulfil together these shared needs?

Outline

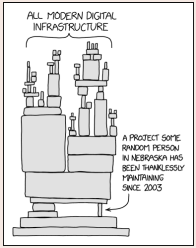
Software supply chain and its issues

Complex digital infrastructure

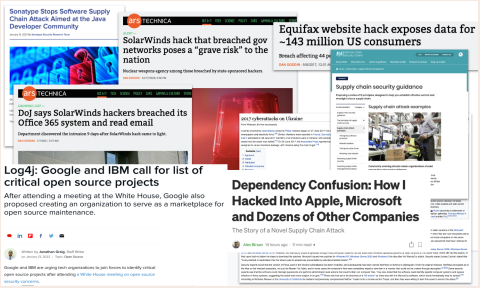


Software supply chain and its issues

Complex digital infrastructure

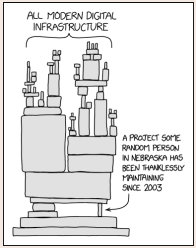


Software supply chain in the news

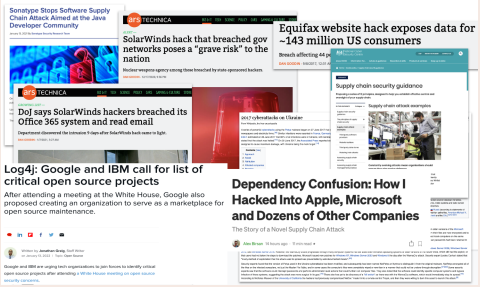


Software supply chain and its issues

Complex digital infrastructure



Software supply chain in the news

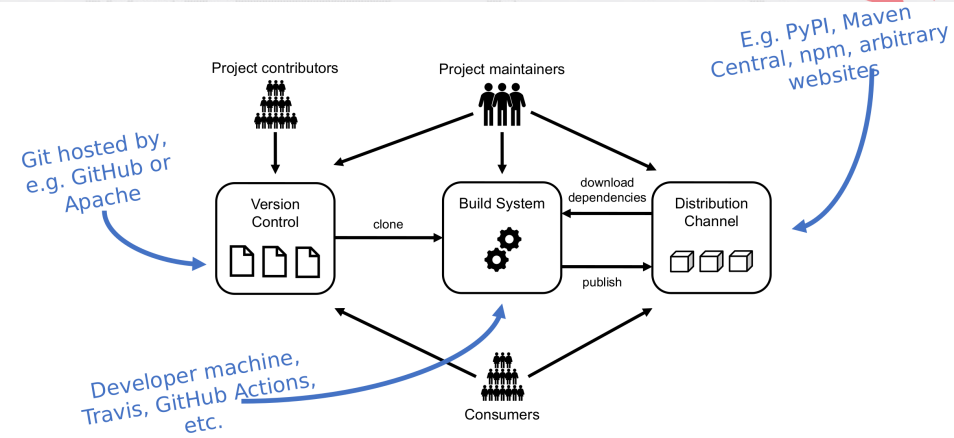


Software Supply Chain attacks

Malicious code injection into software components to compromise downstream users

- March 2022 node-ipc and peacenotwar (CVE-2022-23812)
- Dec 2021 Apache Log4j Remote Code Execution (Log4Shell, CVE-2021-44228)
- Nov 2018 Attack on NPM package event-stream

Software supply chain in a picture



A long road ahead

Vertical approach

improve security of *each component* separately

Horizontal approach

explore *the whole supply chain*

A few key challenging properties

findability	needs qualified metadata
availability	needs an archive and a system of identifiers
integrity	needs crypto
traceability	needs a global provenance database
reproducibility	needs groundbreaking tools

A long road ahead

Vertical approach

improve security of *each component* separately

Horizontal approach

explore *the whole supply chain*

A few key challenging properties

findability	needs qualified metadata
availability	needs an archive and a system of identifiers
integrity	needs crypto
traceability	needs a global provenance database
reproducibility	needs groundbreaking tools

We need a *global coordinated effort…*
and a *common, open, shared* infrastructure to track *all (Open Source) software*!

Forges are key platforms, but they are *not* enough!

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)

Forges are key platforms, but they are *not* enough!

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

Forges are key platforms, but they are *not* enough!

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

Forges are key platforms, but they are *not* enough!

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

source code is spread across hundreds of them…

lack of uniformity, no persistence guarantee

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 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

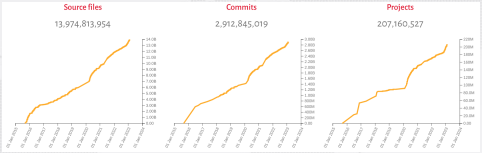
preserve and share all software source code

Research infrastructure

enable analysis of all software source code

One infrastructure
open and shared

One infrastructure
open and shared



Largest archive

One infrastructure
open and shared

Largest archive

Technology

- transparency and FOSS
- replicas all the way down

Content (billions!)

- intrinsic identifiers
- facts and provenance

Organization

- non-profit
- multi-stakeholder

Sharing the vision

UNESCO
United Nations Educational, Scientific and Cultural Organization

And many more ...
www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Inria

Diamond sponsor

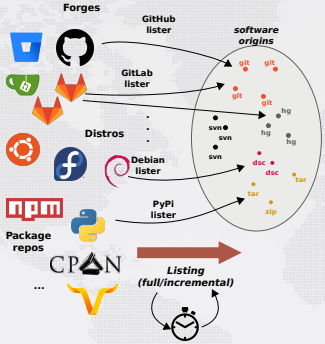
Platinum sponsors

Gold sponsors

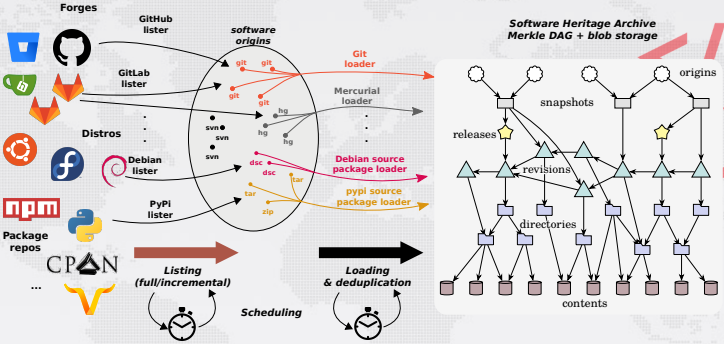
Silver sponsors

Bronze sponsors

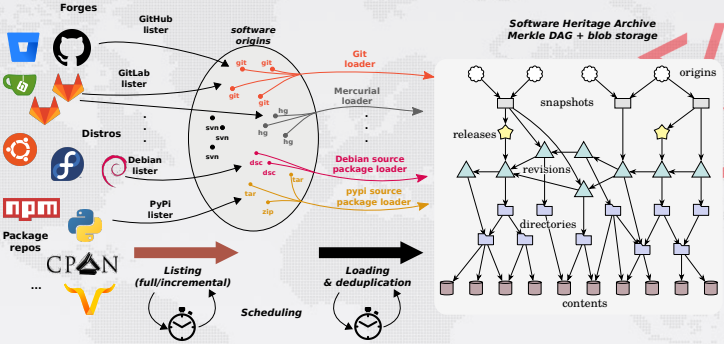
A peek under the hood: a universal archive



A peek under the hood: a universal archive



A peek under the hood: a universal archive



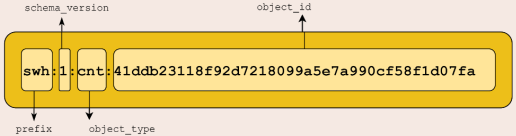
Global development history permanently archived in a uniform data model

- over 14 billion unique source files from over 210 million software projects
- ~1PB (compressed) blobs, ~30 B nodes, ~400 B edges

Software Heritage Identifiers (SWHID)

see swhid.org

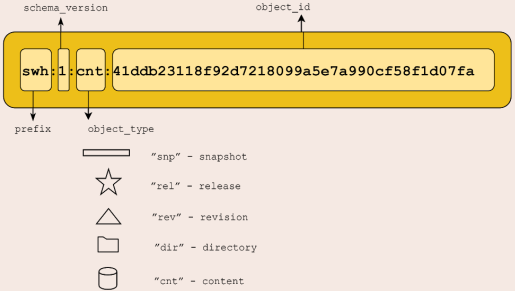
25+B intrinsic, decentralised, cryptographically strong identifiers, SWHIDs



Software Heritage Identifiers (SWHID)

see swhid.org

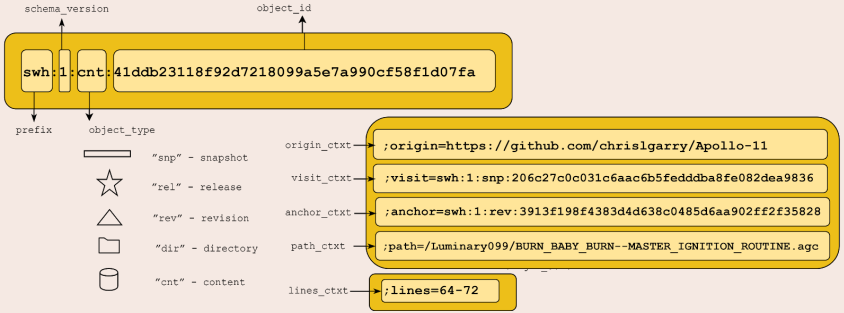
25+B intrinsic, decentralised, cryptographically strong identifiers, SWHIDs



Software Heritage Identifiers (SWHID)

see swhid.org

25+B intrinsic, decentralised, cryptographically strong identifiers, SWHIDs

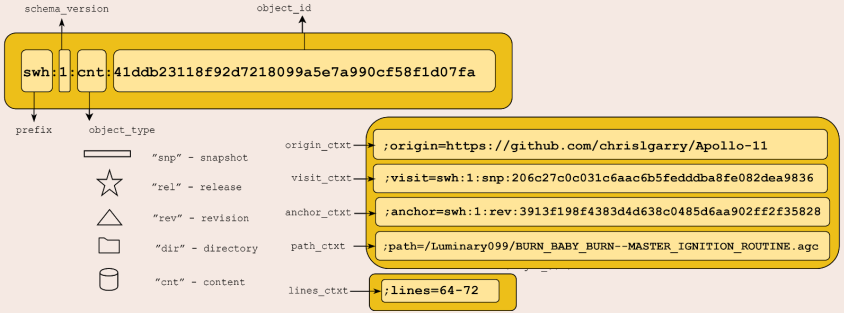


Intrinsic Identifiers for software artefacts

Software Heritage Identifiers (SWHID)

see swhid.org

25+B intrinsic, decentralised, cryptographically strong identifiers, SWHIDs



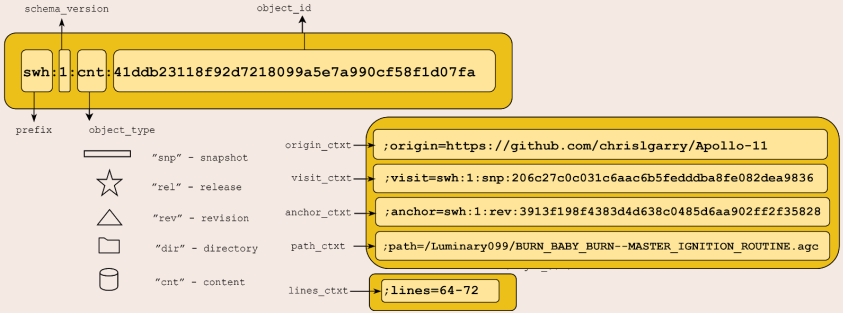
Emerging standard : SPDX 2.2; IANA registered; WikiData P6138; ISO (ongoing)

Intrinsic Identifiers for software artefacts

Software Heritage Identifiers (SWHID)

see swhid.org

25+B intrinsic, decentralised, cryptographically strong identifiers, SWHIDs



Emerging standard : SPDX 2.2; IANA registered; WikiData P6138; ISO (ongoing)

Full fledged *source code references* for reproducibility

Examples: Apollo 11 AGC excerpt, Quake III rsqrt; Guidelines available, see ICMS 2020

A quick tour

- Browse (e.g. Apollo 11, and 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

A revolutionary infrastructure

The *graph* of public software development

All software development in a single graph …

- enable traceability

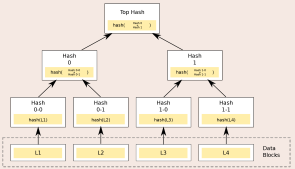
A revolutionary infrastructure

The *graph* of public software development

All software development in a single graph …

- enable traceability

The *global ledger* of public code



… a Merkle graph

- ensure integrity

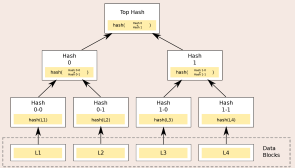
A revolutionary infrastructure

The *graph* of public software development

All software development in a single graph …

- enable traceability

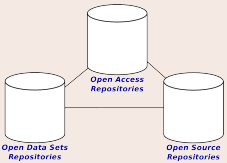
The *global ledger* of public code



… a Merkle graph

- ensure integrity

A *pillar* of Open Science



Reference archive of Research Software

- reproducibility
- reference

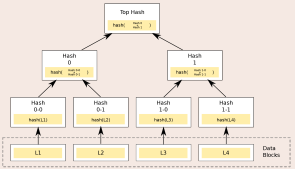
A revolutionary infrastructure

The *graph* of public software development

All software development in a single graph …

- enable traceability

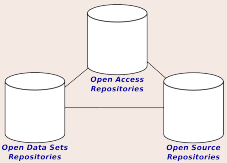
The *global ledger* of public code



… a Merkle graph

- ensure integrity

A *pillar* of Open Science



Reference archive of Research Software

- reproducibility
- reference

Reference platform for *Big Code*

uniform data structure

- large scale studies
- machine learning, AI, …

Industry use cases (selection)

Open Source complete and corresponding source code distribution (Intel)

Software Heritage members can:

- archive source code in Software Heritage, distribute only the SWHID

Industry use cases (selection)

Open Source complete and corresponding source code distribution (Intel)

Software Heritage members can:

- archive source code in Software Heritage, distribute only the SWHID

Traceability and integrity (OIN for the *Linux System Definition*)

Software Heritage members can:

- archive source code in Software Heritage
- track it and verify its integrity using its SWHID

Industry use cases (selection)

Open Source complete and corresponding source code distribution (Intel)

Software Heritage members can:

- archive source code in Software Heritage, distribute only the SWHID

Traceability and integrity (OIN for the *Linux System Definition*)

Software Heritage members can:

- archive source code in Software Heritage
- track it and verify its integrity using its SWHID

And much more!

- an open source, open data code scanner for open compliance (`swh-scanner`)
- security — upcoming PTCC (French CampusCyber) project SWHSec
- supply chain management, long term archive *add your use case here*

Outline

A rally flag for a grand vision

Bring together academia, industry, governments, communities

"to build a reference, global infrastructure for open and better software"

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

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 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

You can help!

fund and/or *develop* SWH, *use* SWH research, *build* tools

softwareheritage.org

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Appendix

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

- Query Software Heritage 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

Source: gitlab.softwareheritage.org/swh/devel/swh-scanner
License: GPL-3+
Package: pypi.org/project/swh.scanner

swh-scanner demo — Efficiency

```
$ du -sh --exclude=.git /srv/src/linux/git
4,1G /srv/src/linux/git
```

```
$ time swh scanner scan /srv/src/linux
Files:                  78277
    known:      78267 ( 99%)
directories:             5085
      fully-known:       5081 ( 99%)
  partially-known:          4 (  0%)

38,65s user 4,71s system 81% cpu 53,127 total
```

```
$ swh scanner scan --output-format ndjson /srv/src/linux/git | grep false
...
{"scripts/kconfig/symbol.o": {"swhid": "swh:1:cnt:874f19...", "known": false}}
...
```

PTCC

Axe 1 : Programme R&D

SWHSec

Leveraging Software Heritage to Enhance Cybersecurity

Co-porteur
Nom : Barais
Prénom : Olivier
Email : olivier.barais@irisa.fr

Co-porteur
Nom : Di Cosmo
Prénom : Roberto
Email : roberto@dicosmo.org

Co-porteur
Nom : Zacchiroli
Prénom : Stefano
Email : stefano.zacchiroli@telecom-paris.fr

1

Enjeux stratégiques du projet

*Aujourd'hui, **seulement les hyperscalers** fournissent les plateformes qui hébergent les codes sources et distribuent les binaires des logiciels Open Source*

- *Github (Microsoft)*
- *Gitlab (gitlab.com et un large ensemble de repositories privés)*

*On assiste à la **concentration des outils d'analyse de la supply chain open source** entre des acteurs **non européens**, e.g.:*

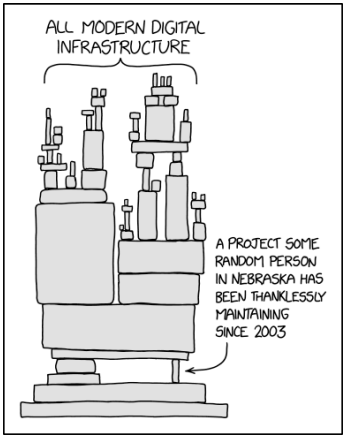
- *Sur les dépendances (Dependatbot https://github.com/dependabot)*
- *Sur le code (rough-auditing-tool-for-security RATS, CodeQL by github, …)*

Opportunité: un outil unique au monde (SWH) qui fournit une source de données vaste et peut être outillé pour la cybersécurité

Les objectifs du projet

Construire par-dessus l'infrastructure de Software Heritage (SWH), une chaîne d'analyse et de remédiation unique scalable dédiée à la cybersécurité

- ***Analyseur du code source**, capable de bénéficier de l'architecture de SWH*
- *Infrastructure de gestion des dépendances permettant une **analyse temporelle de l'évolution des dépendances** dans le domaine de l'open-source*
- *Analyse de l'**impact d'une vulnérabilité** à l'aide de SWH*
- *Remédiation sur un ensemble de projets d'une vulnérabilité découverte*
- ***Extension de Software Heritage** pour la prise en compte des outils d'analyse de sécurité*



10

SWHSec

Leveraging Software Heritage to Enhance Cybersecurity