Software Heritage

Collecting, Preserving and Sharing all the Source Code

Roberto Di Cosmo

roberto@dicosmo.org

May 7th, 2019

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

Short Bio: Roberto Di Cosmo

Computer Science professor in Paris, now working at INRIA

- *30 years* of research (Theor. CS, Programming, Software Engineering, Erdos #: 3)
- *20 years* of Free and Open Source Software
- *10 years* building and directing structures for the common good

1999 *DemoLinux* – first live GNU/Linux distro

2007 *Free Software Thematic Group*
150 members 40 projects 200Me

2015 *Software Heritage* at INRIA

2018 *National Committee for Open Science*, France

Source code: *executable* and *human readable*

"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");
}
```

An essential part of our knowledge

Harold Abelson, Structure and Interpretation of Computer Programs (1985)

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

Len Shustek, Computer History Museum (2006)

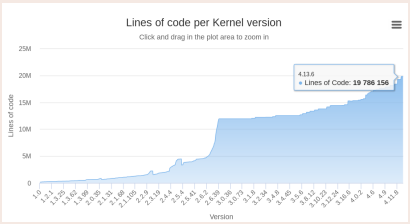
"Source code provides a view into the mind of the designer."

Apollo 11 (~60.000 lines), 1969

"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton

Linux Kernel (in your pockets!)



We are at a turning point

Loosing our heritage

- the key people are passing away
- we miss a catalog
- we miss an archive

Research lagging behind

- research software is undervalued
- software research is under-recognised
- reproducibility is not guaranteed

Generalised lack of awareness

- policy: EU copyright reform, Open Science
- general public: insufficient education

Diversity far from ideal

- "geek" / "silicon valley" culture
- gender and geography

Outline

Software Heritage, in a nutshell

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Collect, preserve and share the *source code* of *all the software*

Preserving our heritage, enabling better software and better science for all

Reference catalog

find and reference all the source code

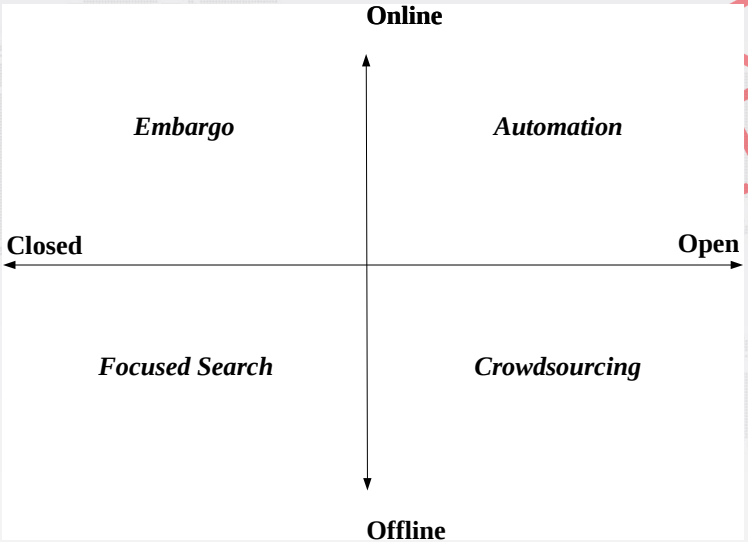
Universal archive

preserve all the source code

Research infrastructure

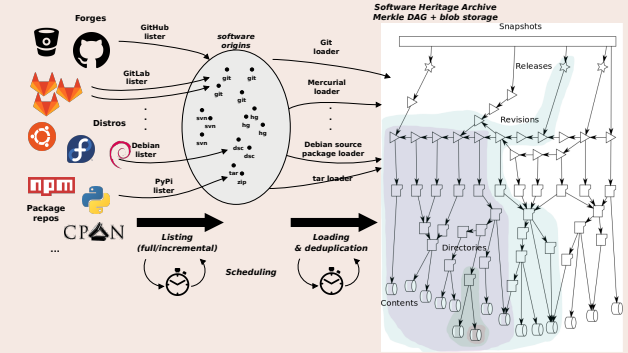
enable analysis of all the source code

All the source code: strategy



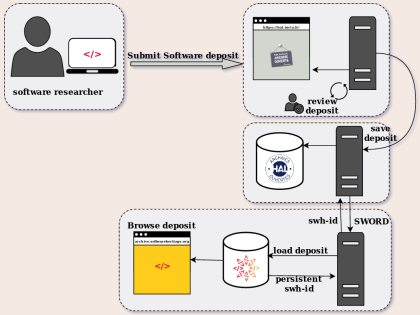
Pull and push

Automation (pull)



We collect everything (no filtering)

Deposit (push)



Curation process in place
for research software (credit)

Outline

Using the archive

Intrinsic identifiers

- see the iPres 2018 article `bit.ly/swhpidpaper`
- example: swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa;lines=53-82

Wayback machine

- find lost source code
- example: Parmap in a 2012 research article (and how it should be)

Research software

moderated deposit via HAL: guidelines for authors and guidelines for moderators

Saving code explicitly

- see Save code now

Outline

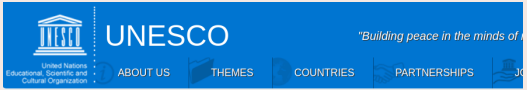
Raising Awareness

April 3rd 2017, Unesco Inria agreement

Inria
INVENTEURS DU MONDE NUMÉRIQUE

UNESCO

November 2018, Unesco Inria expert call



Home › All News › Experts call for greater recognition of software source code as heritage for sustainable development

Experts call for greater recognition of software source code as heritage for sustainable development

16 November 2018

Growing Support

Sharing the vision

Donors, members, sponsors

Outline

You can help!

Collect

- *save source code now*!
- help build a *shared process*
- bootstrap the *focused search*

Preserve

- host *ancillary material*
- join *the SWH Foundation*
- become a *mirror*

Share

- build *on top of* SWH

Connect

- emulation, binaries, metadata, etc…

Awareness

- leverage the Paris Call on Software Source Code (Unesco)

Join the revolution!

Software Heritage

www.softwareheritage.org @swheritage

Library of Alexandria of code

Working together to

- recover the past
- preserve our heritage
- share the knowledge
- prepare the future

Learn more www.softwareheritage.org/publications

Jean-François Abramatic, Roberto Di Cosmo, Stefano Zacchiroli
Building the Universal Archive of Source Code, Communication of the ACM, October 2018

Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli
Identifiers for Digital Objects: the Case of Software Source Code Preservation, iPRES 2018: Intl. Conf. on Digital Preservation

Roberto Di Cosmo, Stefano Zacchiroli
Software Heritage: Why and How to Preserve Software Source Code, iPRES 2017: Intl. Conf. on Digital Preservation

Outline

A principled infrastructure

bit.ly/swhpaper

Technology

- transparency and FOSS
- replicas all the way down

Content

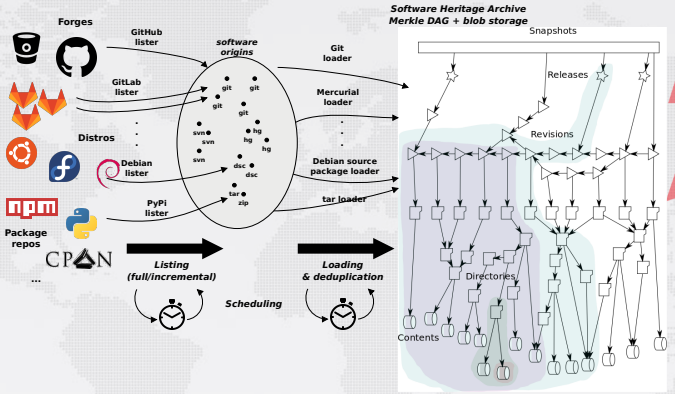
- intrinsic identifiers
- facts and provenance

Organization

- non-profit
- multi-stakeholder

Outline

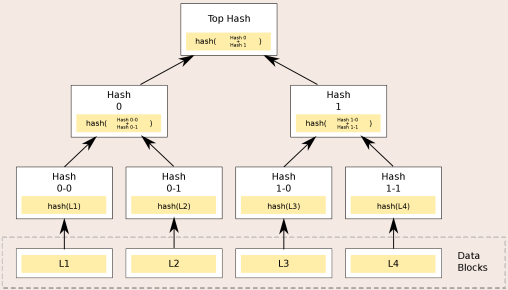
Automation, and storage



- full development history permanently archived
- origins: GitHub (auto), Debian (auto), Gitlab.com, Gitorious, Google Code, GNU
- ~ 200Tb raw contents, ~ 10Tb graph (10Bn nodes, 100Bn edges)

Much more than an archive!

Merkle tree (R. C. Merkle, Crypto 1979)



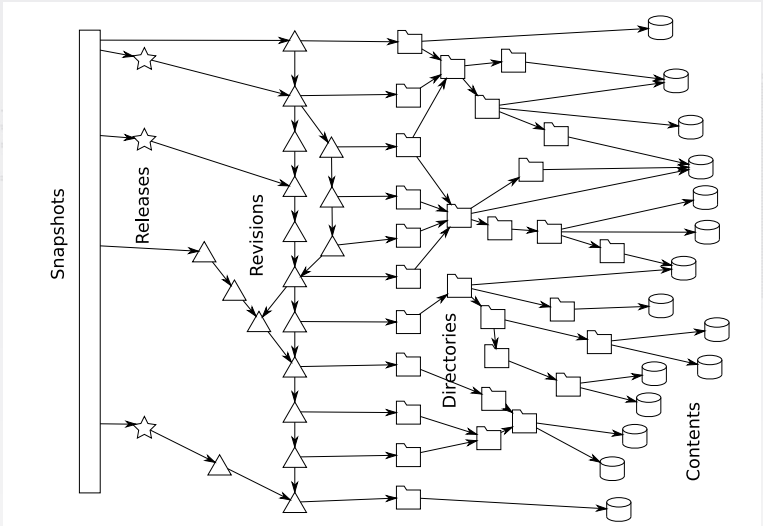
Combination of

- tree
- hash function

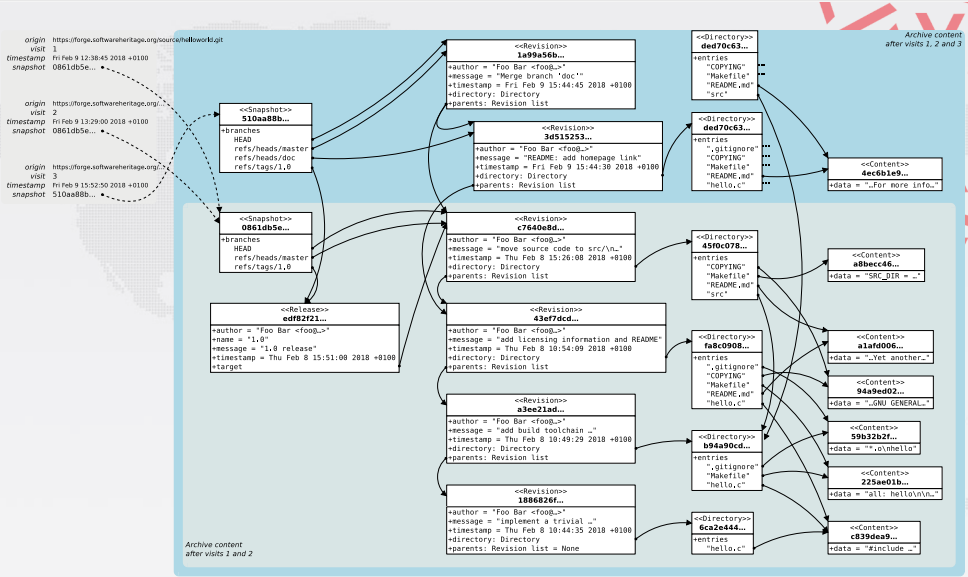
Classical cryptographic construction

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

The archive in pictures



A bird's eye view



Outline

Our challenges in the PID landscape

Typical properties of systems of identifiers

uniqueness, non ambiguity, persistence, abstraction (opacity)

Key needed properties from our use cases

gratis identifiers are free (billions of objects)

integrity the associated object cannot be changed (sw dev, *reproducibility*)

no middle man no central authority is needed (sw dev, *reproducibility*)

we could not find systems with both integrity and no middle man !

An important distinction: DIOs vs. IDOs

The term "Digital Object Identifier" is construed as "digital identifier of an object," rather than "identifier of a digital object" — *Norman Paskin. 2010*

DIO (Digital Identifier of an Object) — identifiers for (potentially) non digital objects

- epistemic complexity (manifestations, versions, locations, etc.)
- need an authority to ensure persistence and uniqueness

IDO (Identifier of a Digital Object) — identifiers (only) for digital objects

- can provide both integrity and no middle man
- broadly used in modern software development (git, etc.)

IDOs and DIOs adress different needs

- for the core Software Heritage IDOs are enough
- we must not use DIOs for reproducibility

The Software Heritage IDO schema (see http://bit.ly/swhpids)

swh:1:**cnt**:94a9ed024d3859793618152ea559a168bbcbb5e2 full text of the GPL3 license

swh:1:**dir**:d198bc9d7a6bcf6db04f476d29314f157507d505 Darktable source code

swh:1:**rev**:309cf2674ee7a0749978cf8265ab91a60aea0f7d

a **revision** in the development history of Darktable

swh:1:**rel**:22ece559cc7cc2364edc5e5593d63ae8bd229f9f

release 2.3.0 of Darktable, dated 24 December 2016

swh:1:**snp**:c7c108084bc0bf3d81436bf980b46e98bd338453

a **snapshot** of the entire Darktable repository (4 May 2017, GitHub)

Current resolvers: archive.softwareheritage.org and n2t.org