

Software Heritage

Building the Universal Software Archive for Open Science

Roberto Di Cosmo

roberto@dicosmo.org

December 12th, 2018



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science
- 4 An inconvenient truth
- 5 The Software Heritage initiative
- 6 A revolutionary infrastructure for Open Source and Open Access
- 7 Under the hood
- 8 Using the Software Heritage archive
- 9 Building for the long term
- 10 Conclusion



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

2008 *Mancoosi project* www.mancoosi.org

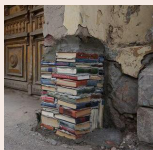
2010 *IRILL* www.irill.org

2015 *Software Heritage* at INRIA

2018 *National Committee for Open Science*, France

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science
- 4 An inconvenient truth
- 5 The Software Heritage initiative
- 6 A revolutionary infrastructure for Open Source and Open Access
- 7 Under the hood
- 8 Using the Software Heritage archive
- 9 Building for the long term
- 10 Conclusion

Key mediator for accessing *all* information (c) Banski

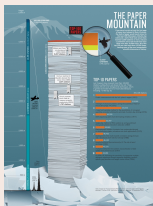


Information is a main pillar of our modern societies.

Absent an ability to correctly interpret digital information, we are left with [...] "rotting bits" [...] of no value.

Vinton G. Cerf IEEE 2011

Software is an essential component of modern scientific research



[...] the vast majority describe experimental methods or software that have become essential in their fields.

Top 100 papers (Nature, October 2014)

Sometimes, if you dont have the software, you dont have the data

Christine Borgman, Paris, 2018

Software embodies our *Knowledge and Cultural Heritage*

The knowledge is in the source code!



"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");
}
```

Harold Abelson, Structure and Interpretation of Computer Programs

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

Quake III source code (excerpt)

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this
    // can be removed

    return y;
}
```

Net. queue in Linux (excerpt)

```
/*
 * SFB uses two B[l][n] : L x N arrays of bins (L levels, N bins per level)
 * This implementation uses L = 8 and N = 16
 * This permits us to split one 32bit hash (provided per packet by rxhash or
 * external classifier) into 8 subhashes of 4 bits.
 */
#define SFB_BUCKET_SHIFT 4
#define SFB_NUMBUCKETS (1 << SFB_BUCKET_SHIFT) /* N bins per Level */
#define SFB_BUCKET_MASK (SFB_NUMBUCKETS - 1)
#define SFB_LEVELS (32 / SFB_BUCKET_SHIFT) /* L */

/* SFB also uses a virtual queue, named "bin" */
struct sfb_bucket {
    u16        qlen; /* length of virtual queue */
    u16        p_mark; /* marking probability */
};
```

Len Shustek, Computer History Museum

“Source code provides a view into the mind of the designer.”

~ 50 years, a lightning fast growth

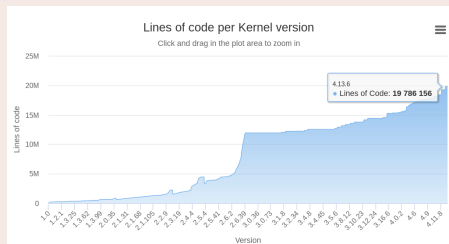
Apollo 11 Guidance Computer (~60.000 lines), 1969



"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



... now in your pockets!

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science**
- 4 An inconvenient truth
- 5 The Software Heritage initiative
- 6 A revolutionary infrastructure for Open Source and Open Access
- 7 Under the hood
- 8 Using the Software Heritage archive
- 9 Building for the long term
- 10 Conclusion



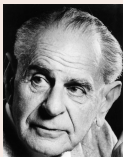
The experimental method



- make an *observation*
- formulate an *hypothesis*
- set up an **experiment**
- elaborate a *theory*

And then we **reproduce** and **verify**.

Reproducibility is the key



non-reproducible single occurrences are of no significance to science

Karl Popper, The Logic of Scientific Discovery, 1934

... evolves in the digital age!

For an experiment involving software, we need

- open access** to the scientific article describing it
- open data sets** used in the experiment
- source code** of all the components
- environment** of execution
- stable references** between all this

Remark

The first two items are already widely discussed!

... what about *software*?

An example from my research field, Computer Science

Analysis of 613 papers

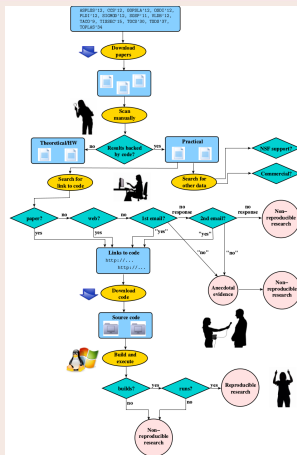
- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

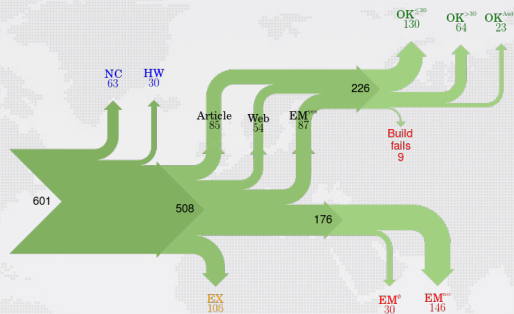
all very practical oriented

The basic question

can we get the code to build and run?

The workflow





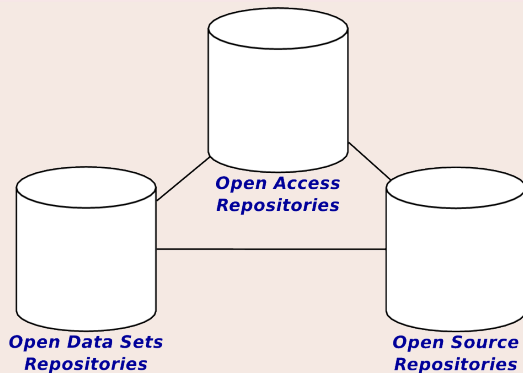
... that's a whopping 40% of **non reproducible** works!

The main reasons

source code (*or the right version of it*) cannot be found

Software Source code is an important pillar

The Magic Triangle of Scientific Knowledge



Nota bene

The links in the picture are **essential**

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science
- 4 An inconvenient truth**
- 5 The Software Heritage initiative
- 6 A revolutionary infrastructure for Open Source and Open Access
- 7 Under the hood
- 8 Using the Software Heritage archive
- 9 Building for the long term
- 10 Conclusion



A forgotten pillar of Open Science

No reference catalog



to find and reference **all**
the source code

No universal archive



to preserve **all** the source
code

No research infrastructure



to enable analysis of **all**
the source code

Lack of recognition

not (yet) a first class citizen

- in the EOSC plan
- in the EU copyright reform
- in the scholarly works

Lack of guidance on how to

- choose a license
- cite a software project
- make source code available
- relate to industry best practices

No catalog, no archive, no references: we are at a turning point

Looking at the past

- a lot of old software misplaced, lost, or behind barriers, but...
- most founding fathers are still here, and willing to share
- **urgent** to collect their knowledge

Only a few years left.

Looking at the future

- software development and use skyrockets: more programmers, and more code!
- **essential** to provide a **universal** platform for all the future software source code

Every year that goes by makes the problem worse.

it is **urgent** to take action!

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science
- 4 An inconvenient truth
- 5 The Software Heritage initiative**
- 6 A revolutionary infrastructure for Open Source and Open Access
- 7 Under the hood
- 8 Using the Software Heritage archive
- 9 Building for the long term
- 10 Conclusion





Software Heritage



Our mission

Collect, preserve and share the source code of all the software that is available

Past, present and future

Preserving the past, enhancing the present, preparing the future



Thomas Jefferson, February 18, 1791

... let us save what remains: not by vaults and locks which fence them from the public eye and use in consigning them to the waste of time, but by such a multiplication of copies, as shall place them beyond the reach of accident.

A *common* infrastructure

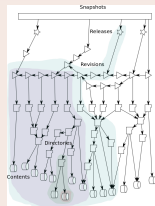
- **mutualisation** for sustainability
- open source, **non for profit**
- mirror network **open to all**
- **may** prevent a useless diaspora

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science
- 4 An inconvenient truth
- 5 The Software Heritage initiative
- 6 A revolutionary infrastructure for Open Source and Open Access**
- 7 Under the hood
- 8 Using the Software Heritage archive
- 9 Building for the long term
- 10 Conclusion



A revolutionary infrastructure for industry

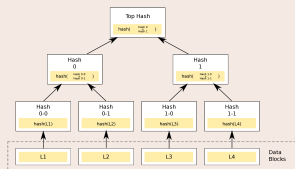
The *graph* of Software Development



All of the software development in **a single graph!**

- **lookup** by content hash
- **wayback machine** for software development
 - <http://archive.softwareheritage.org/>
- ... and much more

The *blockchain* of Software Development

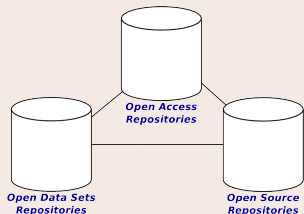


All of a software development... in a single **Merkle** graph!

Widely used crypto (e.g., Git, blockchains, IPFS, ...)

- built-in **deduplication**
- intrinsic, **unforgeable identifiers** at all levels
- simplifies **traceability** (licensing, supply chain management)

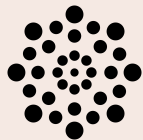
A pillar of Open Science



The *reference archive* of Research Software for **Open Science**

- **curated deposit** of research software
 - in collaboration with **HAL**, **CCSD** and **Inria IES**
 - now open *to all researchers!*
- **intrinsic** identifiers for **reproducibility**

Reference platform for *Big Code*

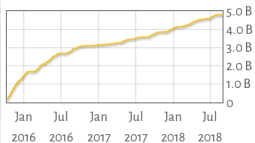


- unique **observatory** of all software development
- **big data, machine learning** paradise: classification, trends, coding patterns, code completion...

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science
- 4 An inconvenient truth
- 5 The Software Heritage initiative
- 6 A revolutionary infrastructure for Open Source and Open Access
- 7 Under the hood**
- 8 Using the Software Heritage archive
- 9 Building for the long term
- 10 Conclusion

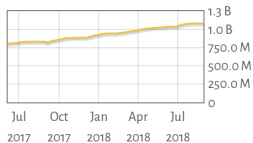
Source files

5,011,613,861



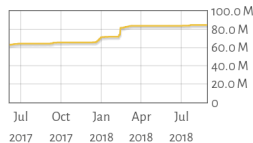
Commits

1,126,348,335



Projects

85,202,432



GitHub

debian



GitLab

Google code



GITORIOUS

GNU

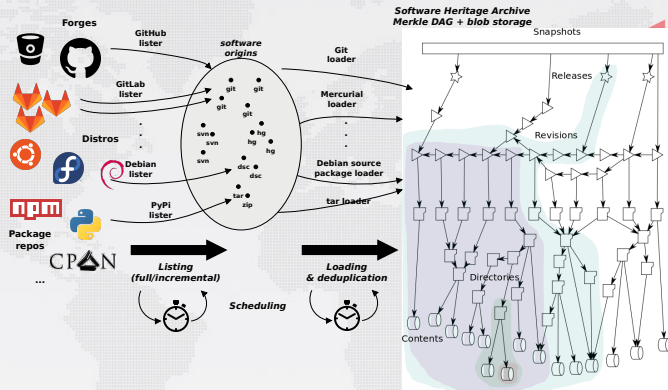
HAL
archives-ouvertes.fr

Inria
inventeurs du monde numérique

python
Package Index

- 200 TB (compressed) blobs, 6 TB database (as a graph: 10 B nodes + 100 B edges)
- The *richest* public source code archive, ... and growing daily!

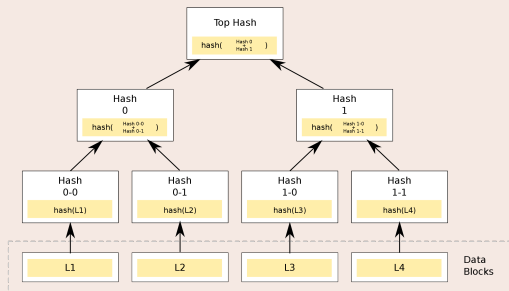
Automation, and storage



- full development history permanently archived

Much more than an archive!

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



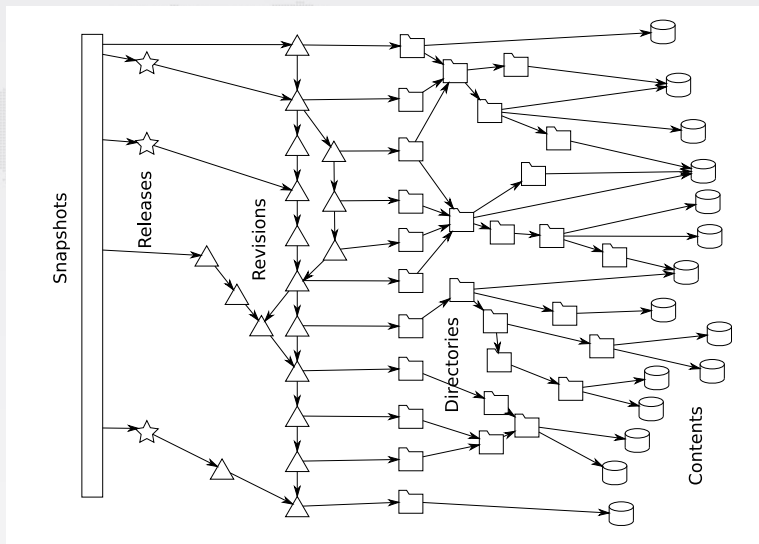
Combination of

- 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



Contents

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

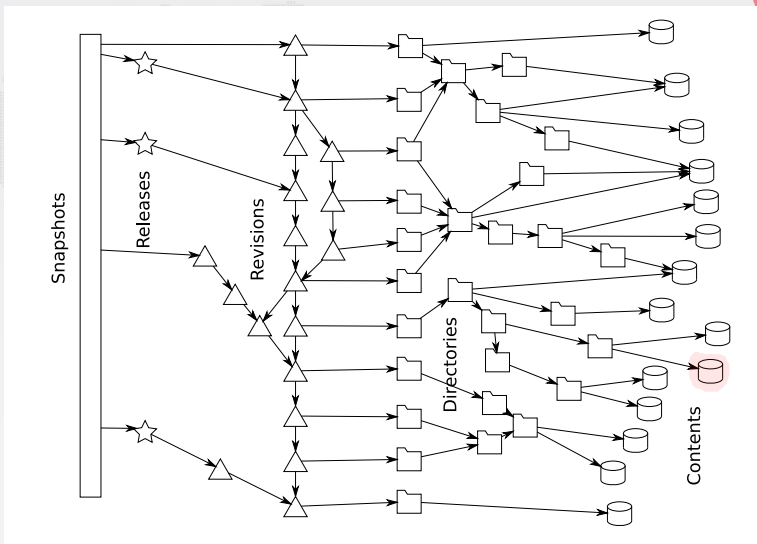
The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program—to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent anyone from denying you
these rights.

sha1: 8624bcdae55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: 94a9ed024d385...
length: 35147

The archive in pictures



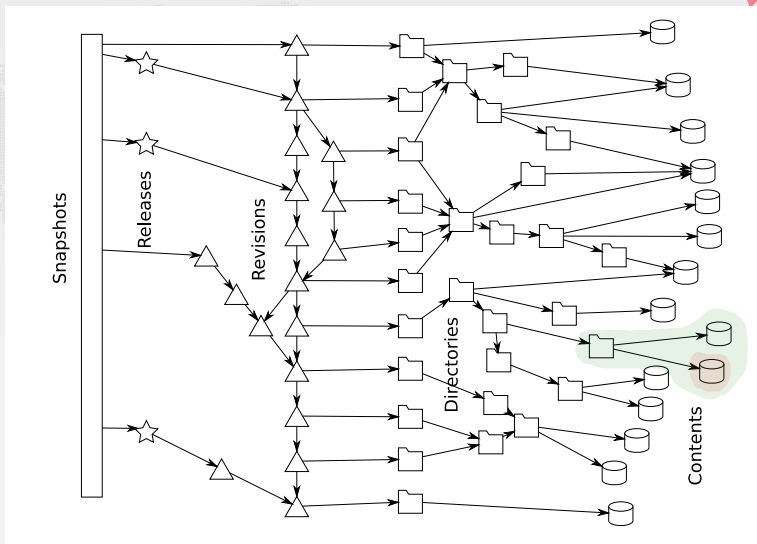


Directories

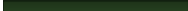
```
100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d0a34af6f52cf3cf6b0c2f7bd0648fbd255e77f AUTHORS
100644 blob 94a9ed024d3859793618152ea559a168bbcbb5e2 LICENSE
100644 blob d9b2665a435a43f8a79a84e0867751dfb095c7bb MANIFEST.in
100644 blob 524175c2bad0b35b975f79284c2f5a6d5eaf2eb4 Makefile
100644 blob 5c7e3a5bbddb038682ba7793f440492ed9678bb3 Makefile.local
100644 blob 8617980629cd24e6080404f09aa749b085b3e07b README.db_testing
100644 blob 76b29f94cf815e0869c414d38d78d7ce08ec514e README.dev
040000 tree e1e10ecfef948af0b93adb0372afc89f12e92618a bin
040000 tree 83e56d0beaf7793c77a45a345c80fcb8af503013 debian
040000 tree a34c9c4ba213f0cedc67f9816348d27955577af5 docs
100644 blob f2a6d32c6135aa7287bbd76167b01df2ae4f1539 requirements.txt
100755 blob eee147c36caf1bbc2d820da8dc026cb5b68180bc setup.py
040000 tree 224bb4c1f4c67fca1d160bffdd2d06094e7e1abf3 sql
040000 tree 8631c9cd77bbe993168107ab5baf51f40c6300be swh
040000 tree 8fb905b56ba8ed692f1209b2773b474c6c1d66c1 utils
```

id: 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d

The archive in pictures



Revisions

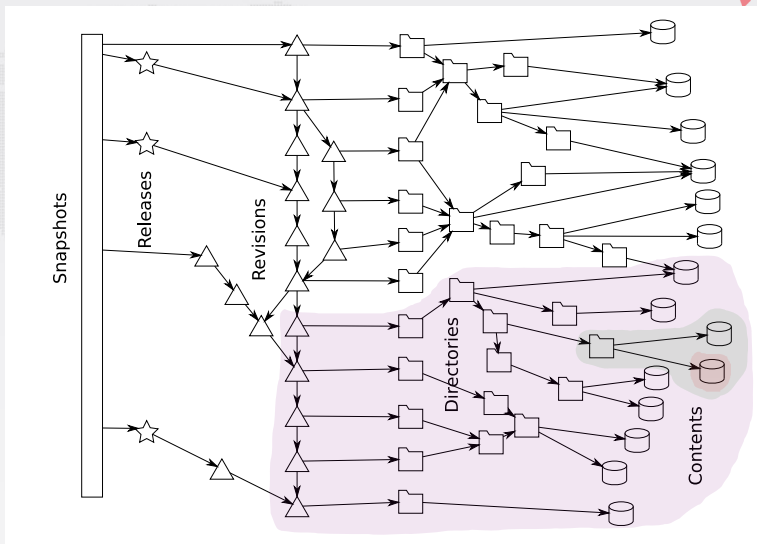
Details	Changes	Files
SHA: 963634dca6ba5dc37e3ee426ba091092c267f9f6		
Author: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Committer: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Subject: provenance.tasks: add the revision -> origin cache task		
Parent: fc3a8b59ca1df424d860f2c29ab07fee4dc35d10 : test...storage: properly pipeline origin and cont...		
provenance.tasks: add the revision -> origin cache task		
sw/h/storage/provenance/tasks.py  77		

tree 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d
parent [fc3a8b59ca1df424d860f2c29ab07fee4dc35d10](#)
author Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200
committer Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200

provenance.tasks: add the revision -> origin cache task

id: 963634dca6ba5dc37e3ee426ba091092c267f9f6

The archive in pictures



Releases

tag v0.0.51
Tagger: Nicolas Dandrimont <nicolas@dandrimont.eu>
Date: Wed Aug 24 14:36:03 2016 +0200

Release sw.h.storage v0.0.51

- Add new metadata column to origin_visit
- Update sw.h-add-directory script for updated API
[...]

commit c0c9f16b1e134f593e7567570a1761b156e6eb1d

object c0c9f16b1e134f593e7567570a1761b156e6eb1d
type commit
tag v0.0.51
tagger Nicolas Dandrimont <nicolas@dandrimont.eu> 1472042163 +0200

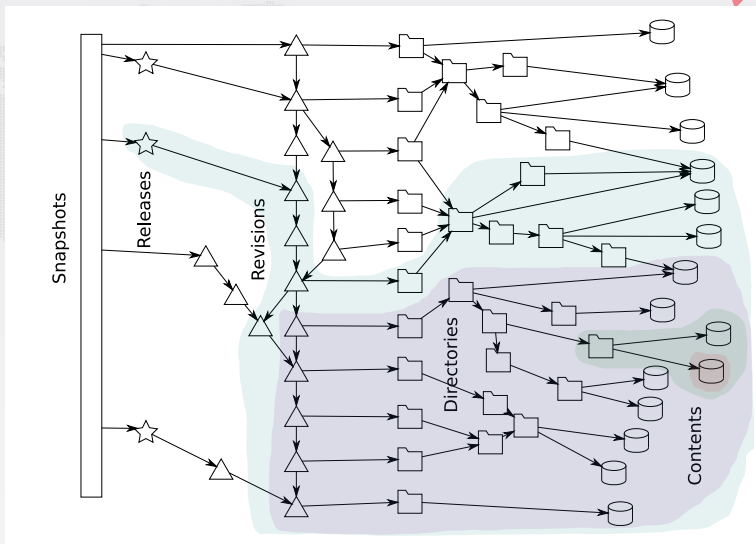
Release sw.h.storage v0.0.51

- Add new metadata column to origin_visit
- Update sw.h-add-directory script for updated API
-----BEGIN PGP SIGNATURE-----

iQIzBAABCAAdBQJXvZTNFhuaWNvbGFzQGRhbmRyaW1vbnQuZXUACgkQ7AWLMO2+
neqorw/aa65Ob5DjzEa+kWN3rXgV5+1K1vEVh1wNKAw8eKJ7aX2kEiLdt7uf
ahpZ6pz3q8nqs6aC1+YrxBfcih3L2YtrdZeWwXqr8xWNMaEoYDb8aaphwh8AD5t2
ICBii2ujtXuCrDt93eKKPwvzXG+h80sMWy35Dr6jW7Z7K4MuPGglyIHPY55yo
IGEndWno7Vfh1Vm6t1n5qB7I5mXRaqA+becqddubTZ2xij+jpLlUqC8cyqN3hm/fL
qsJ2mu8kyz3t8tG/H1/pV+15OwBlNp05TH0tuojoEVgPK/dH5P79QuHDHZFkCao
klj6kAWyU80Mxb+nKVjeLbrR3+yWBFJ3Qp5a1/V8o0Th6E1dALCNMpEaKCoKtMt
d/gMRax111/g0EDfnsW67G6sDwKPKPHngfVLQ3nV3GaQQTnu1RpMz006H9/tAwzC
Gg/K1PdHT4hz0I46wYPZje0U2VXGFu6vVU9vFQ4ZR/Wjn+0zZdcRdrJ5UOMn
RpTTRfUbsXUeXHGOpkgXhSYTnvp1gdPc76U5TsK0aGe84AZm1lk0mGrwXCvFPqYo
nhhibB5HBNMoqyF6yTSOpUbyK70tpYRRUGKwDeRK0wKSxkWKUZGtKzy6jYqJ29
gulwgZQif5qWQC80ontAL2+HvFfaVyckMejUhg62cP/+EHlvUk=
=kOxP
-----END PGP SIGNATURE-----

id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb

The archive in pictures



Snapshots

git show-refs

```
commit 08ffeb25770109525eb3ce21691466c53a1d9158 refs/heads/atime
commit ba5443a24e3f9fe323a46c292cec4fcbe61c67eb refs/heads/directory-listing-arrays
commit d69e0dbf892383ff6589b27fbc1c05d27238d9c5 refs/heads/foo
commit cf7ff9eea0eb22f78946908f5a8019f67de468e08 refs/heads/master
commit 7eca197fc66d2024047e54b1ed9e8b44361a0fc2 refs/heads/tmp-directory-add
commit 642a205f37de85005a85d427b53ee4fb2252e82e refs/heads/tmp/generic-releases
tag 20f043b1379cf768d966597799fd4907c757f755 refs/tags/v0.0.1
tag 72a21991a384e539996dbb867bfb0bee72aee2cd refs/tags/v0.0.10
tag 3590e0ca0ebb070e5b376705fa230bbfa4ffa5cc refs/tags/v0.0.11
tag 33378427a403ba569a67777b8d58f6674fbc6556 refs/tags/v0.0.12
tag 06f74652755b327cf590311c2bfa036cf3b4b35d refs/tags/v0.0.13
tag 5a6325fe86ab854b581d7442667d92a11e32f3bd refs/tags/v0.0.14
tag 586fba4e580b4f5fab05f599367643cbcb1a9c7f refs/tags/v0.0.15
tag 8cd8b885f4098bf363177742bd289f660e5be51c refs/tags/v0.0.16
tag a542444ee3f0fbed35efb202fee035c809abc7d6 refs/tags/v0.0.17
tag 228a2f1650dd1222e556559462e1e06fc4993d9 refs/tags/v0.0.18
tag 606979a4ca05d497fc0d24aad00dce82636ef47c refs/tags/v0.0.19
tag 32bf5a59fc2a323baa6d5f15a6ad5382ec275a67 refs/tags/v0.0.2
tag 3147c3d31ec46cf6492f881e908b1237ebdff2c7 refs/tags/v0.0.20
tag 215ea50daball1e082e0b72e76eb4b6073a87908 refs/tags/v0.0.21
tag 3fb168c2072a5d6252124257a1e5dfc0f5ffa1df refs/tags/v0.0.22
tag 8cdbee8da4d73fc5d262789e460a16ac3c72aba4 refs/tags/v0.0.23
...
```

id: b464cad1b66fff266a37b46ea6e7a04b545e904b

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science
- 4 An inconvenient truth
- 5 The Software Heritage initiative
- 6 A revolutionary infrastructure for Open Source and Open Access
- 7 Under the hood
- 8 Using the Software Heritage archive**
- 9 Building for the long term
- 10 Conclusion

Reference archive for all software

A "wayback machine" for software source code ...

with **intrinsic identifiers!**

- <http://archive.softwareheritage.org/browse>
- <http://bit.ly/swhpids> for persistent identifiers

Demo time: let's highlight some features...

Origin search

Software Heritage Search Help Tools Browse

Search software heritage origins to browse

Origin type	Origin identifier	Attributes
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓
git	github.com:softwareheritage/origin-search	✓

Directory browsing

Software Heritage Search Help Tools Browse

SW/h/origin-search: https://github.com/lythe/origin-search

SW/h/origin-search Directory

File	Meta	Size	Mime type
git.sh	-----	6.0k	text/x-sh; charset=utf-8
Dockerfile	-----	1.7k	text/plain
LICENSE	-----	1.0k	text/plain
README.md	-----	1.0k	text/markdown
src	-----	1.0k	text/plain
test	-----	1.0k	text/plain
tools	-----	1.0k	text/plain

Revisions as diffs

Software Heritage Search Help Tools Browse

SW/h/origin-search

Diff between revisions: sw/h/origin-search/src/git.sh

```
diff --git a/sw/h/origin-search/src/git.sh b/sw/h/origin-search/src/git.sh
index 1234567..8901234
+ #!/bin/sh
+ set -e
+ # This script is used to bootstrap the origin-search repository.
+ # It will clone the repository and set up the environment.
+ # Usage: git.sh [options]
+ # Options:
+ # -h, --help            display this help message
+ # -v, --verbose          show verbose output
+ # -s, --source           source repository URL
+ # -d, --destination     destination repository URL
+ # -u, --update           update the repository
+ # -c, --clone            clone the repository
+ # -m, --mirror           mirror the repository
+ # -n, --no-checkout      do not checkout the repository
+ # -o, --origin           origin repository name
+ # -b, --branch           branch to checkout
+ # -f, --force            force the update
+ # -k, --key              ssh key path
+ # -p, --port             ssh port
+ # -P, --proxy            proxy server
+ # -S, --ssh-command     ssh command
+ # -T, --timeout          timeout in seconds
+ # -U, --url              repository URL
+ # -W, --workdir          work directory
+ # -X, --extra            extra options
+ # --help                display this help message
+ --
+ if [ "$#" = 0 ]; then
+     echo "Usage: git.sh [options]"
+     exit 1
+ fi
+ while [ "$#" -gt 0 ]; do
+     case "$1" in
+         -h | --help)
+             echo "display this help message"
+             exit 0
+         ;;
+         -v | --verbose)
+             set -x
+         ;;
+         -s | --source)
+             source="$2"
+             shift
+         ;;
+         -d | --destination)
+             destination="$2"
+             shift
+         ;;
+         -u | --update)
+             update="true"
+         ;;
+         -c | --clone)
+             clone="true"
+         ;;
+         -m | --mirror)
+             mirror="true"
+         ;;
+         -n | --no-checkout)
+             no_checkout="true"
+         ;;
+         -o | --origin)
+             origin="$2"
+             shift
+         ;;
+         -b | --branch)
+             branch="$2"
+             shift
+         ;;
+         -f | --force)
+             force="true"
+         ;;
+         -k | --key)
+             key="$2"
+             shift
+         ;;
+         -p | --port)
+             port="$2"
+             shift
+         ;;
+         -P | --proxy)
+             proxy="$2"
+             shift
+         ;;
+         -S | --ssh-command)
+             ssh_command="$2"
+             shift
+         ;;
+         -T | --timeout)
+             timeout="$2"
+             shift
+         ;;
+         -U | --url)
+             url="$2"
+             shift
+         ;;
+         -W | --workdir)
+             workdir="$2"
+             shift
+         ;;
+         -X | --extra)
+             extra="$2"
+             shift
+         ;;
+         *)
+             echo "Unknown option: $1"
+             exit 1
+         ;;
+     esac
+ done
+ if [ -z "$source" ]; then
+     echo "Source repository URL is required"
+     exit 1
+ fi
+ if [ -z "$destination" ]; then
+     echo "Destination repository URL is required"
+     exit 1
+ fi
+ if [ -z "$origin" ]; then
+     echo "Origin repository name is required"
+     exit 1
+ fi
+ if [ -z "$branch" ]; then
+     echo "Branch to checkout is required"
+     exit 1
+ fi
+ if [ -z "$url" ]; then
+     echo "Repository URL is required"
+     exit 1
+ fi
+ if [ -z "$workdir" ]; then
+     echo "Work directory is required"
+     exit 1
+ fi
+ if [ -z "$ssh_command" ]; then
+     echo "SSH command is required"
+     exit 1
+ fi
+ if [ -z "$timeout" ]; then
+     echo "Timeout in seconds is required"
+     exit 1
+ fi
+ if [ -z "$extra" ]; then
+     echo "Extra options are required"
+     exit 1
+ fi
+ # Clone the repository
+ git clone --branch=$branch --depth=1 --single-branch --no-checkout $source $workdir
+ # Set the origin
+ cd $workdir
+ git remote add origin $url
+ # Checkout the branch
+ git checkout $branch
+ # Update the repository
+ git pull --force $origin $branch
+ # Mirror the repository
+ git mirror $origin $workdir
+ # Push the repository
+ git push --force $origin $branch
```

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science
- 4 An inconvenient truth
- 5 The Software Heritage initiative
- 6 A revolutionary infrastructure for Open Source and Open Access
- 7 Under the hood
- 8 Using the Software Heritage archive
- 9 Building for the long term**
- 10 Conclusion

April 3rd 2017, Unesco Inria agreement

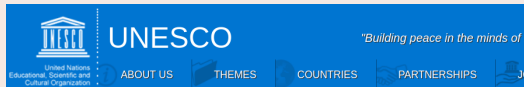
Inria
INVENTEURS DU MONDE NUMÉRIQUE



Roberto Di Cosmo

www.dicosmo.org

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



(CC-BY 4.0) www.softwareheritage.org

December 12th, 2018

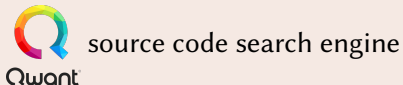
23 / 26

Growing Support

Sharing the vision



Research collaboration



Donors, members, sponsors



Global network

- first independent mirror
- increased reliability

You can help!

Many scientific and technological challenges

object storage, machine learning, classification, efficient graph queries, mirror protocols,
...

Contribute

- `forge.softwareheritage.org`

Funding

- become a partner/sponsor/mirror :
`sponsorship.softwareheritage.org`
- give *your own contribution* :
`www.softwareheritage.org/donate`

Spread the word!

- *use* the archive and help others do
- tell everybody about Software Heritage

- 1 Introductions
- 2 Software is Knowledge...
- 3 Software source code: a pillar of Open Science
- 4 An inconvenient truth
- 5 The Software Heritage initiative
- 6 A revolutionary infrastructure for Open Source and Open Access
- 7 Under the hood
- 8 Using the Software Heritage archive
- 9 Building for the long term
- 10 Conclusion



Software Heritage



www.softwareheritage.org

@swheritage

Library of Alexandria of code

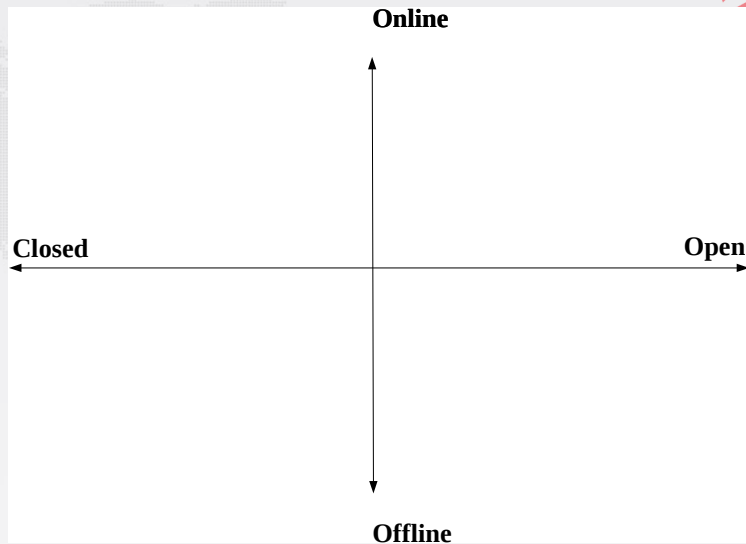


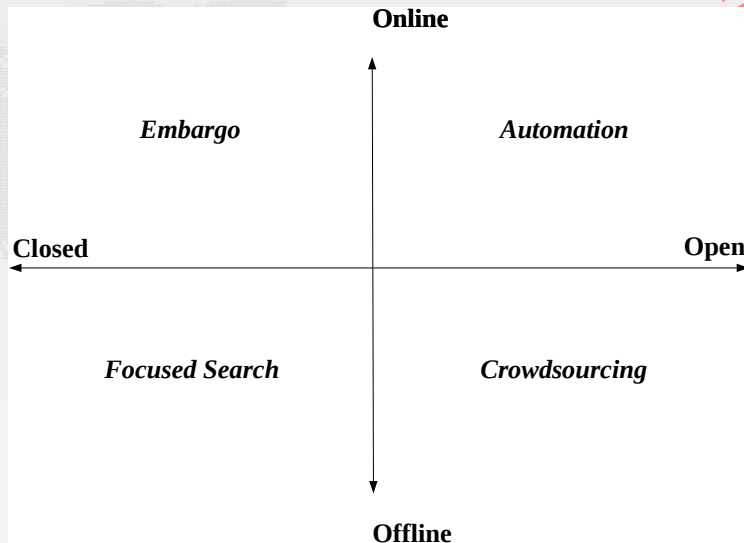
- recover the past
- structure the future

A CERN for Software

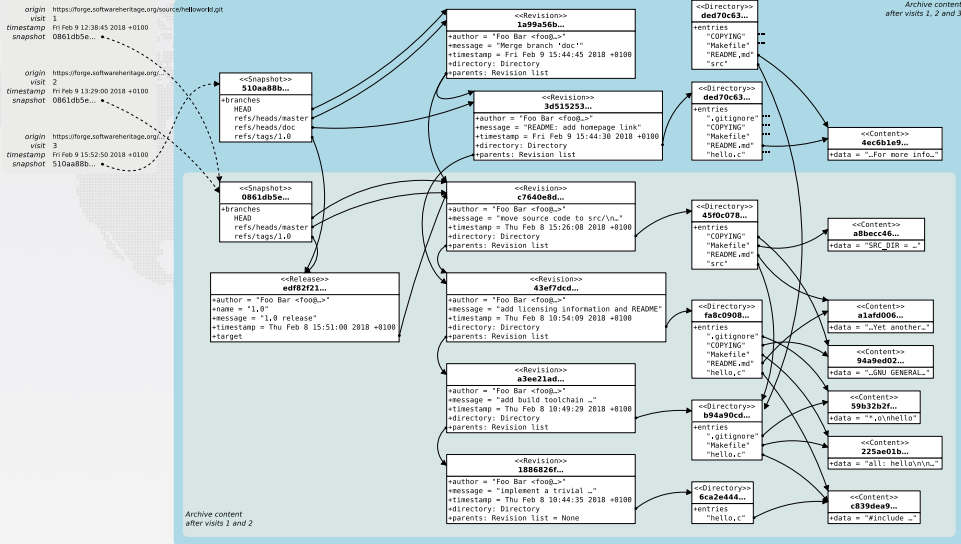


- build better software
 - for industry
 - for society as a whole





A bird's eye view





11 Looking for the right identifiers

A system of identifiers is

- a set of labels (the identifiers)
- mechanisms to perform :

<i>Generation (minting)</i>	create a new label
<i>Assignment</i>	associate label to object
<i>Retrieval</i>	get object from a label

- optionally, mechanisms to perform:

<i>Verification</i>	check label and object
<i>Reverse Lookup</i>	get label from an object
<i>Description</i>	get metadata of an object

Mechanisms offered in some systems of identifiers



Mech. / System	Handle	DOI	Ark	PURL
Generation	Yes	Yes	Yes	Yes
Assignment	Yes	Yes	Yes	Yes
Retrieval	Yes	Yes	Yes	Yes
Verification	N.A.	N.A.	N.A.	N.A.
Reverse Lookup	N.A.	N.A.	N.A.	N.A.
Description	Yes	Yes	Yes	N.A.

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)

digital 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)

digital identifiers (only) for **digital objects**

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

for the core Software Heritage archive, **IDs are enough**

