Software Heritage

Building the Universal Software Archive for Open Science

Roberto Di Cosmo

roberto@dicosmo.org

October 17th, 2018

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

1. Introductions
2. Software is everywhere…
3. … and we are not taking care of it!
4. The Software Heritage initiative
5. Looking for the right PIDs
6. PIDs analysis
7. Using the Software Heritage archive
8. Open Science
9. Building for the long term
10. Conclusion

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

2008 *Mancoosi project* `www.mancoosi.org`

2010 *IRILL* `www.irill.org`

2015 *Software Heritage* at INRIA

2018 *National Committee for Open Science*, France

Outline

1. Introductions
2. Software is everywhere…
3. … and we are not taking care of it!
4. The Software Heritage initiative
5. Looking for the right PIDs
6. PIDs analysis
7. Using the Software Heritage archive
8. Open Science
9. Building for the long term
10. Conclusion

Software is everywhere

Source code is *executable* and *human readable* knowledge

a growing part of our *Cultural Heritage*

Source code is *special*

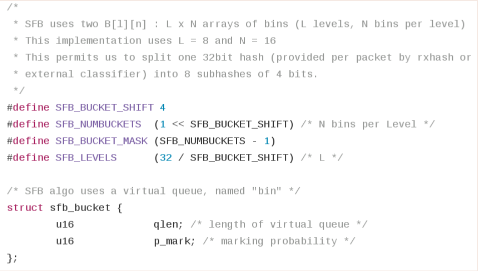
Harold Abelson, Structure and Interpretation of Computer Programs

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

Quake III source code (excerpt)

```c
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)



```c
/*
 * 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 algo 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.”

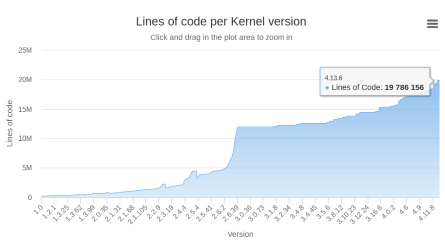
~ 50 years, a lightning fast growth

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!

are we taking care of all this?

Outline

1. Introductions
2. Software is everywhere…
3. … and we are not taking care of it!
4. The Software Heritage initiative
5. Looking for the right PIDs
6. PIDs analysis
7. Using the Software Heritage archive
8. Open Science
9. Building for the long term
10. Conclusion

Software is spread all around

Sourceforge
Debian
CPAN
Gitorious
Maven
Inria
Bitbucket
GitHub
BerliOs
CTAN
Gitlab
CRAN
GoogleCode
Adullact

Software is fragile

damage
disaster
malicious
media
obsolete
attack
aging
tear
dependencies
storage
reference
deletion
dangling
wear
corruption
encryption
format

Software lacks its own research infrastructure

Photo: ALMA(ESO/NAOJ/NRAO), R. Hills

Research software: a long way to go!

ICSE (Zannier, Melrik, Maurer, 2006)

- complete absence of replication studies

ACM TOSEM 2001 to 2006 C. Ghezzi `http://bit.ly/tosemreprod`

- 60% of all papers have tools: only 20% *installable*

Collberg's 2015 study `http://reproducibility.cs.arizona.edu/`

- 601 mainstream papers: 508 with tools, only 40% *installable*

Main reasons

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

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!

Outline

1 Introductions

2 Software is everywhere…

3 … and we are not taking care of it!

4 The Software Heritage initiative

5 Looking for the right PIDs

6 PIDs analysis

7 Using the Software Heritage archive

8 Open Science

9 Building for the long term

10 Conclusion

Software Heritage

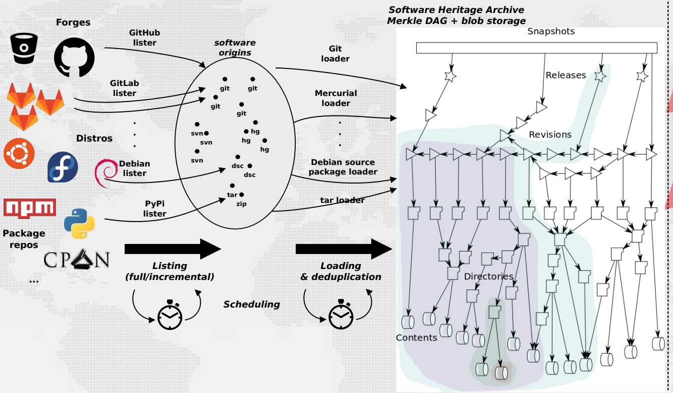
Our mission

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

Past, present and future

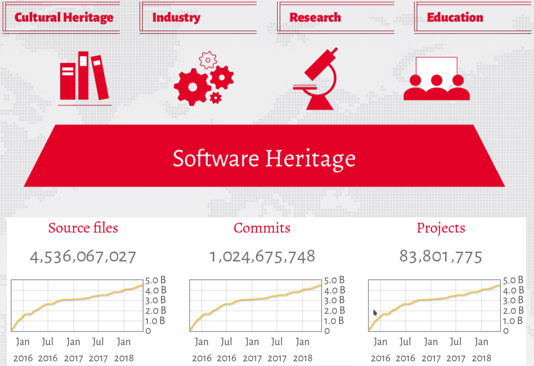
Preserving the past, *enhancing* the present, *preparing* the future

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)

A principled infrastructure http://bit.ly/swhpaper



Cultural Heritage

Industry

Research

Education

Software Heritage

Source files

4,536,067,027

Commits

1,024,675,748

Projects

83,801,775

Technology

- transparency and FOSS
- replicas all the way down

Content

- intrinsic identifiers
- facts and provenance

Organization

- non-profit
- mirror network

Outline

1. Introductions
2. Software is everywhere…
3. … and we are not taking care of it!
4. The Software Heritage initiative
5. Looking for the right PIDs
6. PIDs analysis
7. Using the Software Heritage archive
8. Open Science
9. Building for the long term
10. Conclusion

URL decay disrupts the *web of reference*

Web links *are not* permanent (even *permalinks*)

there is no general guarantee that a URL... which at one time points to a given object continues to do so

T. Berners-Lee et al. Uniform Resource Locators. RFC 1738.

URLs used in articles *decay*!

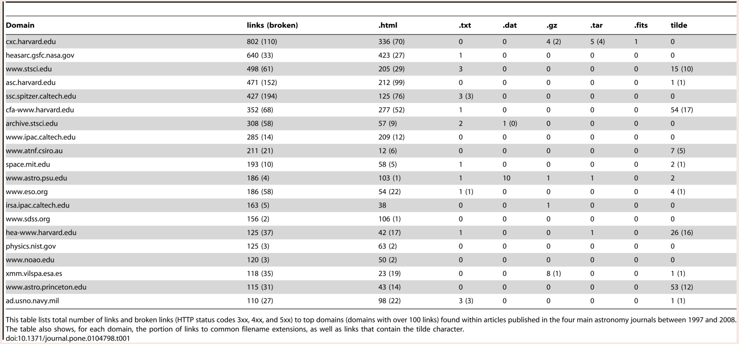
Analysis of *IEEE Computer* (Computer), and the *Communications of the ACM* (CACM): 1995-1999

- the *half-life* of a referenced URL *is approximately 4 years* from its publication date
  D. Spinellis. The Decay and Failures of URL References.
  Communications of the ACM, 46(1):71-77, January 2003.

Similar findings in Lawrence, S. et al. *Persistence of Web References in Scientific Research*, IEEE Computer, 34(2), pp. 26–31, 2001.

Scholar roster of broken links

An example from Astronomy



Domain	links (broken)	.html	.txt	.dat	.gz	.tar	.fits	tilde
cxc.harvard.edu	802 (110)	336 (70)	0	0	4 (2)	5 (4)	1	0
heasarc.gsfc.nasa.gov	640 (33)	423 (27)	1	0	0	0	0	0
www.stsci.edu	498 (61)	205 (29)	3	0	0	0	0	15 (10)
asc.harvard.edu	471 (152)	212 (99)	0	0	0	0	0	1 (1)
ssc.spitzer.caltech.edu	427 (194)	125 (76)	3 (3)	0	0	0	0	0
cfa-www.harvard.edu	352 (68)	277 (52)	1	0	0	0	0	54 (17)
archive.stsci.edu	308 (58)	57 (9)	2	1 (0)	0	0	0	0
www.ipac.caltech.edu	285 (14)	209 (12)	0	0	0	0	0	0
www.atnf.csiro.au	211 (21)	12 (6)	0	0	0	0	0	7 (5)
space.mit.edu	193 (10)	58 (5)	1	0	0	0	0	2 (1)
www.astro.psu.edu	186 (4)	103 (1)	1	10	1	1	0	2
www.eso.org	186 (58)	54 (22)	1 (1)	0	0	0	0	4 (1)
irsa.ipac.caltech.edu	163 (5)	38	0	0	1	0	0	0
www.sdss.org	156 (2)	106 (1)	0	0	0	0	0	0
hea-www.harvard.edu	125 (37)	42 (17)	1	0	0	1	0	26 (16)
physics.nist.gov	125 (3)	63 (2)	0	0	0	0	0	0
www.noao.edu	120 (3)	50 (2)	0	0	0	0	0	0
xmm.vilspa.esa.es	118 (35)	23 (19)	0	0	8 (1)	0	0	1 (1)
www.astro.princeton.edu	115 (31)	43 (14)	0	0	0	0	0	53 (12)
ad.usno.navy.mil	110 (27)	98 (22)	3 (3)	0	0	0	0	1 (1)

This table lists total number of links and broken links (HTTP status codes 3xx, 4xx, and 5xx) to top domains (domains with over 100 links) found within articles published in the four main astronomy journals between 1997 and 2008. The table also shows, for each domain, the portion of links to common filename extensions, as well as links that contain the tilde character.

doi:10.1371/journal.pone.0104798.t001

How Do Astronomers Share Data?

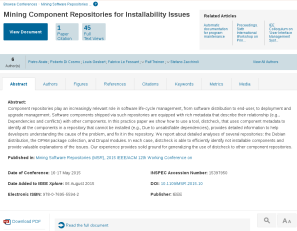
Pepe, Goodman, Muench, Crosas, Erdmann *PLOS August 28, 2014*

dx.doi.org/10.1371/journal.pone.0104798

DOI limitations

Example: doi:10.1109/MSR.2015.10

- to find what 10.1109/MSR.2015.10 is, go to a *resolver* (e.g. doi.org)
- this returns `http://ieeexplore.ieee.org/document/7180064/`
- at this URL we find ...


Browse Conferences > Mining Software Repositories ...
View Document
1 Paper Citation
45 Full Text Views
Related Articles
Automatic documentation for program maintenance
Proceedings. Sixth International Workshop on Prin...
IEE Colloquium on 'User Interface Management Syst...
6 Author(s)
View All Authors
Abstract
Authors
Figures
References
Citations
Keywords
Metrics
Media
Download PDF
Read the full document
AA

Mining Component Repositories for Installability Issues

Pietro Abate ; Roberto Di Cosmo ; Louis Gesbert ; Fabrice Le Fessant ; Ralf Treinen ; Stefano Zacchiroli

Abstract:
Component repositories play an increasingly relevant role in software life-cycle management, from software distribution to end-user, to deployment and upgrade management. Software components shipped via such repositories are equipped with rich metadata that describe their relationship (e.g., Dependencies and conflicts) with other components. In this practice paper we show how to use a tool, distcheck, that uses component metadata to identify all the components in a repository that cannot be installed (e.g., Due to unsatisfiable dependencies), provides detailed information to help developers understanding the cause of the problem, and fix it in the repository. We report about detailed analyses of several repositories: the Debian distribution, the OPAM package collection, and Drupal modules. In each case, distcheck is able to efficiently identify not installable components and provide valuable explanations of the issues. Our experience provides solid ground for generalizing the use of distcheck to other component repositories.

Published in: Mining Software Repositories (MSR), 2015 IEEE/ACM 12th Working Conference on

Date of Conference: 16-17 May 2015

INSPEC Accession Number: 15397950

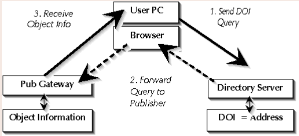
Date Added to IEEE *Xplore*: 06 August 2015

DOI: 10.1109/MSR.2015.10

Electronic ISBN: 978-0-7695-5594-2

Publisher: IEEE

Architecture of the DOI infrastructure



- DOI resolution *can change*
- content at URL *can change*
- no *intrinsic* way of noticing
- persistence based on *good will* of *multiple parties*

Outline

1. Introductions
2. Software is everywhere…
3. … and we are not taking care of it!
4. The Software Heritage initiative
5. Looking for the right PIDs
6. PIDs analysis
7. Using the Software Heritage archive
8. Open Science
9. Building for the long term
10. Conclusion

Systems of identifiers

A *system of identifiers* is

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

Generation (minting)	create a new label
Assignment	associate label to object
Retrieval	get object from a label

- optionally, mechanisms to perform:

Verification	check label and object
Reverse Lookup	get label from an object
Description	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)

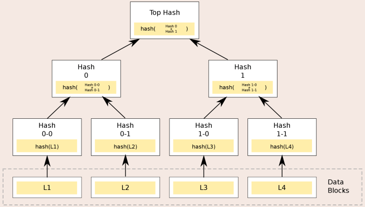
digital identifiers (only) for digital objects

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

for the core Software Heritage archive, IDOs are enough

IDOs in Software Development: the origins

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



Combination of

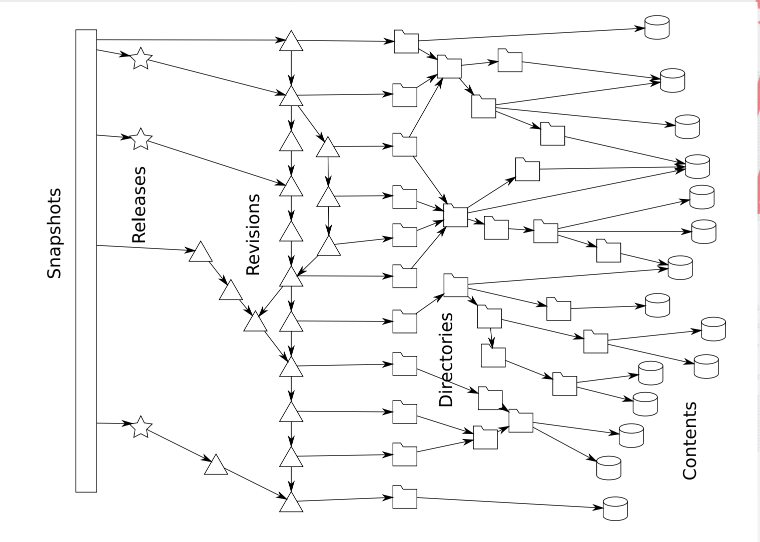
- tree
- hash function

Classical cryptographic construction

fast, parallel signature of large data structures, built-in deduplication

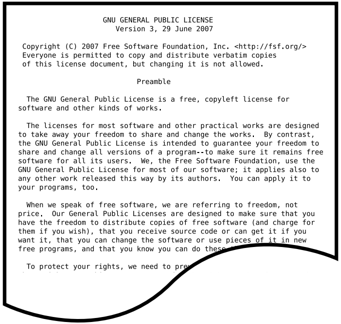
- satisfies all three criteria: gratis, integrity, no middle man!
- widely used in industry (e.g., Git, nix, blockchains, IPFS, ...)

IDOs in Software Heritage: a worked example



IDOs in Software Heritage: a worked example

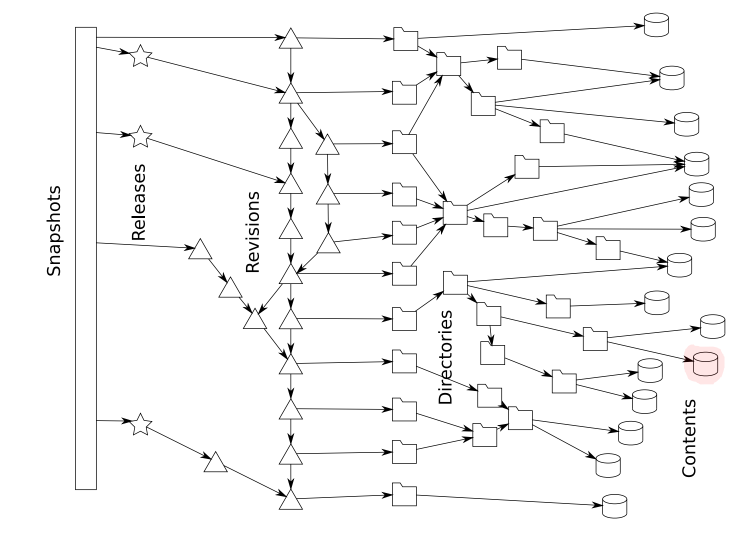
Contents



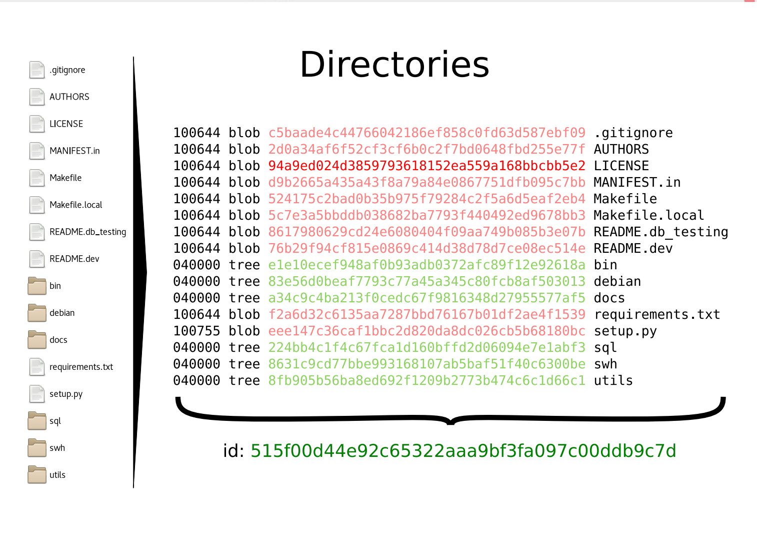
```
                       GNU GENERAL PUBLIC LICENSE
                          Version 3, 29 June 2007

 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

  To protect your rights, we need to pre
```

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

IDOs in Software Heritage: a worked example



IDOs in Software Heritage: a worked example



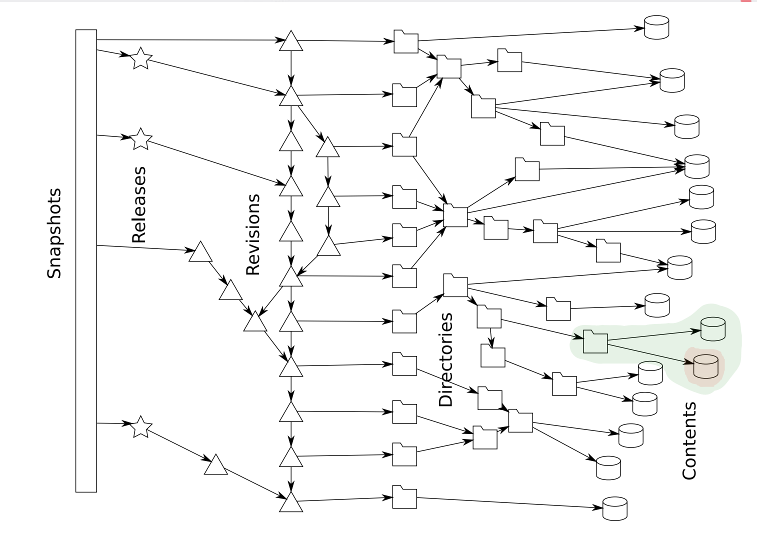
Directories

.gitignore
AUTHORS
LICENSE
MANIFEST.in
Makefile
Makefile.local
README.db_testing
README.dev
bin
debian
docs
requirements.txt
setup.py
sql
swh
utils

```
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 e1e10ecef948af0b93adb0372afc89f12e92618a bin
040000 tree 83e56d0beaf7793c77a45a345c80fcb8af503013 debian
040000 tree a34c9c4ba213f0cedc67f9816348d27955577af5 docs
100644 blob f2a6d32c6135aa7287bbd76167b01df2ae4f1539 requirements.txt
100755 blob eee147c36caf1bbc2d820da8dc026cb5b68180bc setup.py
040000 tree 224bb4c1f4c67fca1d160bffd2d06094e7e1abf3 sql
040000 tree 8631c9cd77bbe993168107ab5baf51f40c6300be swh
040000 tree 8fb905b56ba8ed692f1209b2773b474c6c1d66c1 utils
```

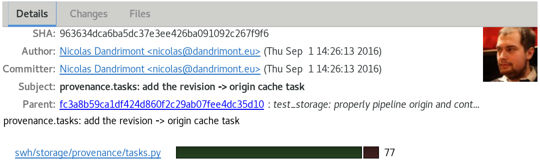
id: 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d

IDOs in Software Heritage: a worked example



IDOs in Software Heritage: a worked example

Revisions



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...*

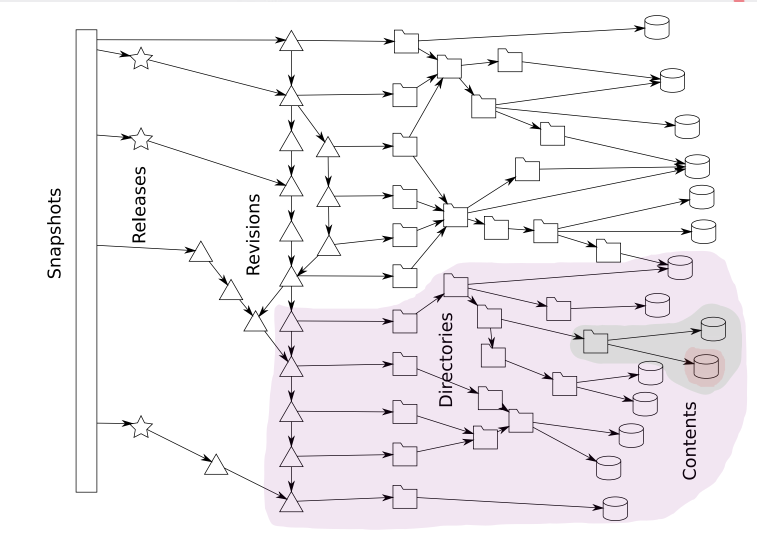
provenance.tasks: add the revision -> origin cache task

swh/storage/provenance/tasks.py 77

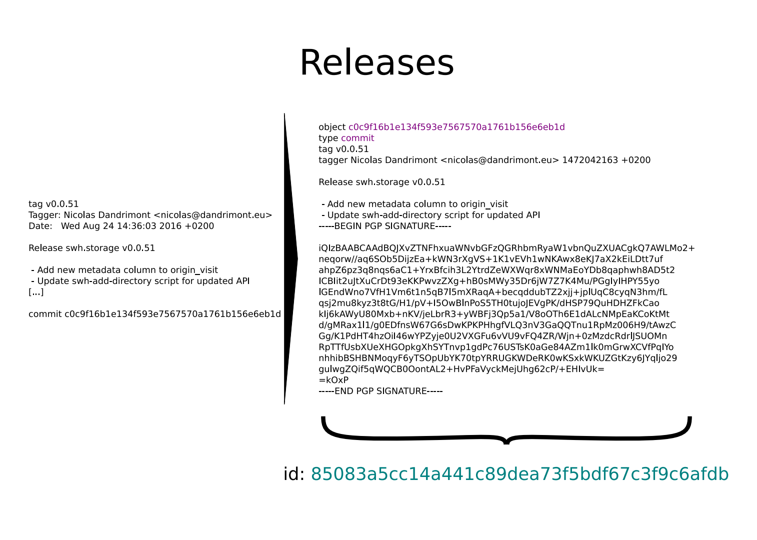
```
tree 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d
parent fc3a8b59ca1df424d860f2c29ab07fee4dc35d10
author Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200
committer Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200

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

id: 963634dca6ba5dc37e3ee426ba091092c267f9f6

IDOs in Software Heritage: a worked example



IDOs in Software Heritage: a worked example



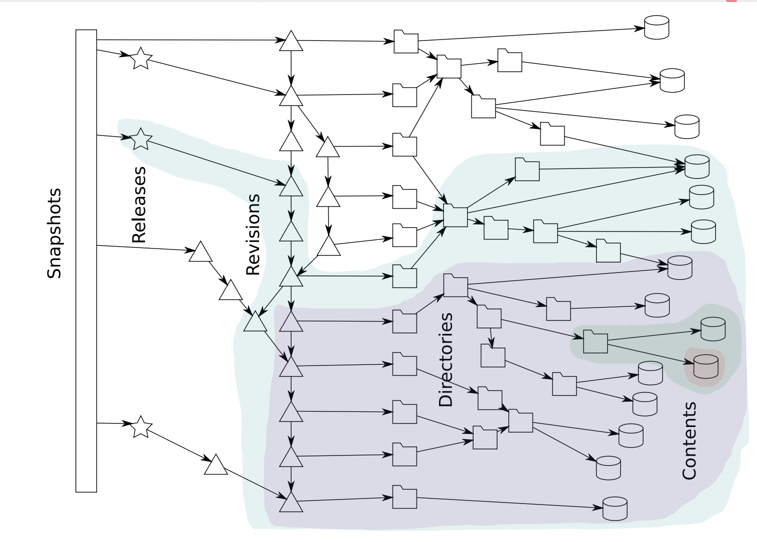
Releases

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

Release swh.storage v0.0.51

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

commit c0c9f16b1e134f593e7567570a1761b156e6eb1d
```

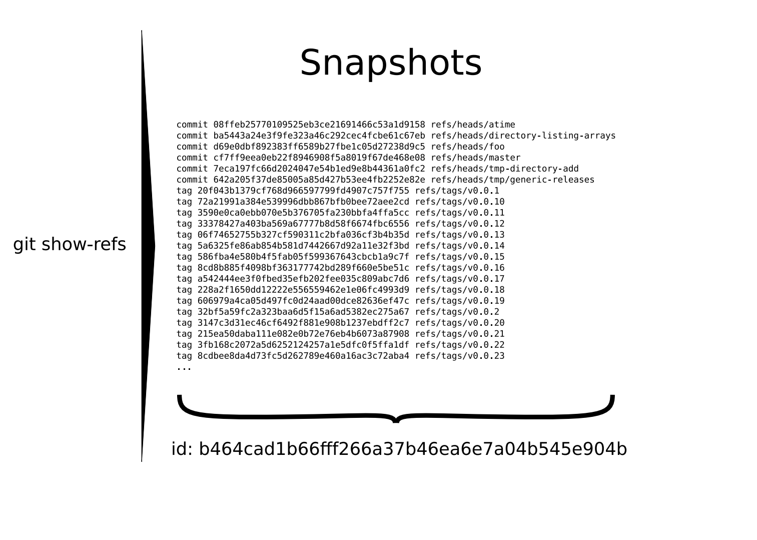
```
object c0c9f16b1e134f593e7567570a1761b156e6eb1d
type commit
tag v0.0.51
tagger Nicolas Dandrimont <nicolas@dandrimont.eu> 1472042163 +0200

Release swh.storage v0.0.51

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

iQIzBAABCAAdBQJXvZTNFhxuaWNvbGFzQGRhbmRyaW1vbnQuZXUACgkQ7AWLMo2+
neqorw//aq6SOb5DijzEa+kWN3rXgVS+1K1vEVh1wNKAwx8eKJ7aX2kEiLDtt7uf
ahpZ6pz3q8nqs6aC1+YrxBfcih3L2YtrdZeWXWqr8xWNMaEoYDb8qaphwh8AD5t2
ICBIit2uJtXuCrDt93eKKPwvzZXg+hB0sMWy35Dr6jW7Z7K4Mu/PGgIyIHPY55yo
lGEndWno7VfH1Vm6t1n5qB7l5mXRaqA+becqddubTZ2xjj+jplUqC8cyqN3hm/fL
qsj2mu8kyz3t8tG/H1/pV+I5OwBlnPoS5TH0tujoJEVgPK/dHSP79QuHDHZFkCao
kIj6kAWyU80Mxb+nKV/jeLbrR3+yWBFj3Qp5a1/V8oOTh6E1dALcNMpEaKCoKtMt
d/gMRax1l1/g0EDfnsW67G6sDwKPKPHhgfVLQ3nV3GaQQTnu1RpMz006H9/tAwzC
Gg/K1PdHT4hzOiI46wYPZyje0U2VXGFu6vVU9vFQ4ZR/Wjn+0zMzdcRdrlJSUOMn
RpTTfUsbXUeXHGOpkgXhSYTnvp1gdPc76USTsK0aGe84AZm1lk0mGrwXCVfPqIYo
nhhibBSHBNMoqyF6yTSOpUbYK70tpYRRUGKWDeRK0wKSxkWKUZGtKzy6JYqIjo29
gulwgZQif5qWQCB0OontAL2+HvPFaVyckMejUhg62cP/+EHIvUk=
=kOxP
-----END PGP SIGNATURE-----
```

id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb

IDOs in Software Heritage: a worked example



IDOs in Software Heritage: a worked example



Snapshots

git show-refs

```
commit 08ffeb25770109525eb3ce21691466c53a1d9158 refs/heads/atime
commit ba5443a24e3f9fe323a46c292cec4fcbe61c67eb refs/heads/directory-listing-arrays
commit d69e0dbf892383ff6589b27fbe1c05d27238d9c5 refs/heads/foo
commit cf7ff9eea0eb22f8946908f5a8019f67de468e08 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 228a2f1650dd12222e556559462e1e06fc4993d9 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 215ea50daba111e082e0b72e76eb4b6073a87908 refs/tags/v0.0.21
tag 3fb168c2072a5d6252124257a1e5dfc0f5ffa1df refs/tags/v0.0.22
tag 8cdbee8da4d73fc5d262789e460a16ac3c72aba4 refs/tags/v0.0.23
...
```

id: b464cad1b66fff266a37b46ea6e7a04b545e904b

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`

Outline

1. Introductions
2. Software is everywhere…
3. … and we are not taking care of it!
4. The Software Heritage initiative
5. Looking for the right PIDs
6. PIDs analysis
7. Using the Software Heritage archive
8. Open Science
9. Building for the long term
10. Conclusion

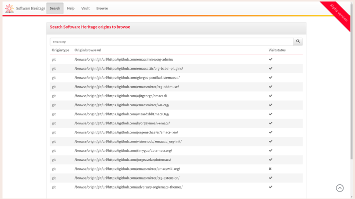
Reference archive for all software

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

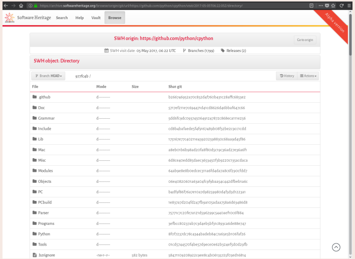


Search Software Heritage origins to browse

emacs org

Origin type	Origin browse url	Visit status
git	/browse/origin/git/url/https://github.com/emacsimize/org-admin/	✓
git	/browse/origin/git/url/https://github.com/emacsattic/org-babel-plugins/	✓
git	/browse/origin/git/url/https://github.com/giorgos-pontikakis/emacs.d/	✓
git	/browse/origin/git/url/https://github.com/emacsmirror/org-oddmuse/	✓
git	/browse/origin/git/url/https://github.com/ajitgeorge/emacs.d/	✓
git	/browse/origin/git/url/https://github.com/emacsmirror/wn-org/	✓
git	/browse/origin/git/url/https://github.com/wizardxbl/EmacsOrg/	✓
git	/browse/origin/git/url/https://github.com/byorgey/noah-emacs/	✓
git	/browse/origin/git/url/https://github.com/jorgenschaefer/emacs-ixio/	✓
git	/browse/origin/git/url/https://github.com/visionnoob/.emacs.d_org-init/	✓
git	/browse/origin/git/url/https://github.com/timyguo/dotemacs.org/	✓
git	/browse/origin/git/url/https://github.com/jorgeavelar/dotemacs/	✓
git	/browse/origin/git/url/https://github.com/emacsmirror/emacswiki.org/	✗
git	/browse/origin/git/url/https://github.com/emacsmirror/org-extension/	✓
git	/browse/origin/git/url/https://github.com/adversary-org/emacs-themes/	✓

Directory browsing



https://archive.softwareheritage.org/browse/origin/git/url/https://github.com/python/cpython/visit/2017-05-05T06:22:05Z/directory/

SWH origin: https://github.com/python/cpython

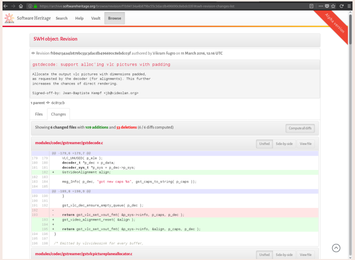
SWH visit date: 05 May 2017, 06:22 UTC   Branches (1799)   Releases (2)

SWH object: Directory

Branch: HEAD   977fc4b /

File	Mode	Size	Sha1 git
.github	d---------		b266746952a70c852d4f760b431c28affc6853e2
Doc	d---------		5717ef211e7c694471d41cd8626d46bbaf647c66
Grammar	d---------		5dd6fc9dc0937450649124787 2c868eca111e256
Include	d---------		cd8b4bafaede5f4f9167489b08f32be2c9cc1cdd
Lib	d---------		175167e77c40211e439202598830c68aa9d45f86
Mac	d---------		a8eb01b6b98ad20fa8f80d5c19c36ad27e36a6f1
Misc	d---------		6d8ce40edd85daec3653450f3b9220c135acdaca
Modules	d---------		64ab9e8e8b0edcec311a6fd4da7a8c6f290cfdd7
Objects	d---------		06e40820601a63a04fc9f9baa54c442dfbeb1a6c
PC	d---------		b4df9f86f76a7e10a7d98259980d4f5d5d1223a1
PCbuild	d---------		1e837a7d204fd247fb9a105adaa758a6d63486d8
Parser	d---------		75771c7c20fe7a121d596299c5440aef10c6f884
Programs	d---------		3efbcc80237ab7c3d4eb5bf31c893ca6de88e747
Python	d---------		8f0f7237dc78c4344badeb84c7a65e5b106faf26
Tools	d---------		01cd5744570f4be57d9ece0e62b324ef5d0d29fb
.bzrignore	-rw-r--r--	582 bytes	58471109208922c9ee8c4b06135725f03ed16814

Revisions as diffs



https://archive.softwareheritage.org/browse/revision/f1b94134a4b879bc55c3dacdb496690c8ebdc03f/#swh-revision-changes-list

SWH object: Revision

Revision f1b94134a4b879bc55c3dacdb496690c8ebdc03f authored by Vikram Fugro on 11 March 2016, 12:16 UTC

```
gstdecode: support alloc'ing vlc pictures with padding

Allocate the output vlc pictures with dimensions padded,
as requested by the decoder (for alignments). This further
increases the chances of direct rendering.

Signed-off-by: Jean-Baptiste Kempf <jb@videolan.org>
```

1 parent 6c813cb

Showing 6 changed files with 109 additions and 53 deletions (6 / 6 diffs computed)

modules/codec/gstreamer/gstdecode.c

```
@@ -179,6 +179,7 @@
     VLC_UNUSED( p_ele );
     decoder_t *p_dec = p_data;
     decoder_sys_t *p_sys = p_dec->p_sys;
+    GstVideoAlignment align;

     msg_Info( p_dec, "got new caps %s", gst_caps_to_string( p_caps ));

@@ -189,8 +190,9 @@
     }

     gst_vlc_dec_ensure_empty_queue( p_dec );
-
-    return gst_vlc_set_vout_fmt( &p_sys->vinfo, p_caps, p_dec );
+    gst_video_alignment_reset( &align );
+
+    return gst_vlc_set_vout_fmt( &p_sys->vinfo, &align, p_caps, p_dec );
 }

 /* Emitted by vlcvideosink for every buffer,
```

modules/codec/gstreamer/gstvlcpictureplaneallocator.c

Outline

1. Introductions
2. Software is everywhere. . .
3. . . . and we are not taking care of it!
4. The Software Heritage initiative
5. Looking for the right PIDs
6. PIDs analysis
7. Using the Software Heritage archive
8. Open Science
9. Building for the long term
10. Conclusion

The way forward to archive and reference scientific software

Scientific software portals

- curate metadata (software deposit, software citation, DIOs)
- archive deposited source code in Software Heritage
- obtain persistent intrinsic IDOs (integrity, not dependent on resolvers, see iPres2018 article `http://bit.ly/swhpidpaper`) linking into Software Heritage

Benefits of connecting with Software Heritage

Features all those of Software Heritage *for free*

Now browse, download

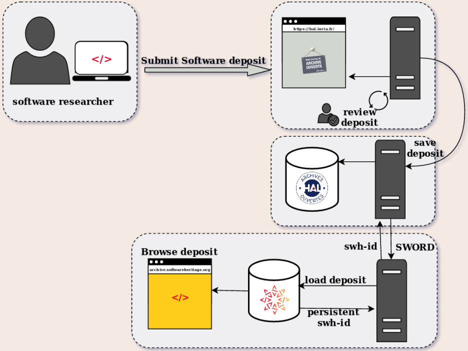
WIP metadata, licenses, provenance (plagiarism detection), classification, ...

Coverage and Uniformity

- one archive for all domains (industry included)
- reference *any* software, not just the deposited ones
- git-compatible identifiers greatly simplify workflows

Deposit Scientific Software

Deposit software in HAL `http://hal.inria.fr/hal-01738741`



Generic mechanism:

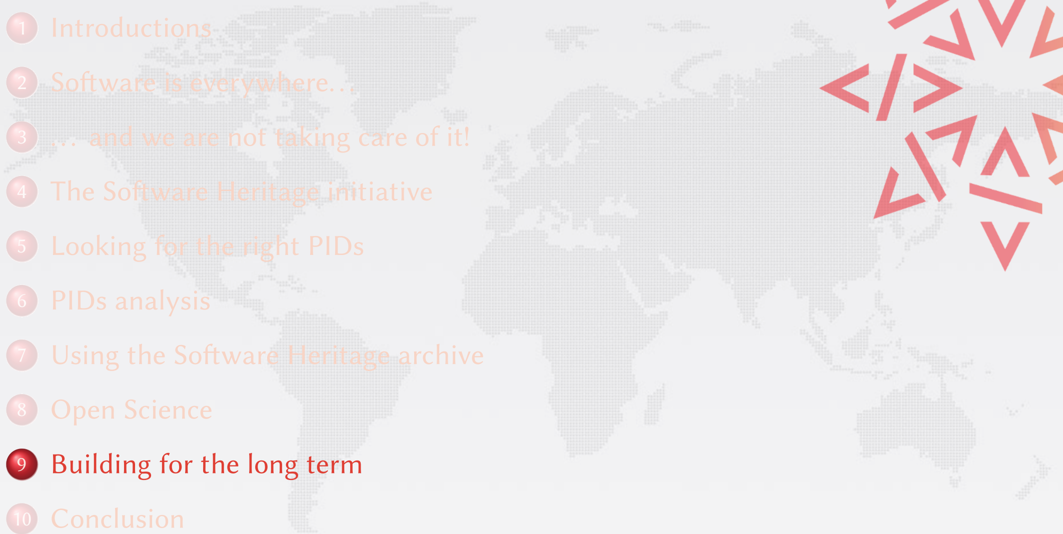
- SWORD based
- review process
- versioning

How to do it:

- today: deposit .zip or .tar.gz file (*guide*)
- tomorrow:
  - provide *SWH id* and metadata
  - include *metadata file* for automatic metadata extraction
  - ...

September 2018: open to all on `https://hal.archives-ouvertes.fr/`

Outline



1. Introductions
2. Software is everywhere…
3. … and we are not taking care of it!
4. The Software Heritage initiative
5. Looking for the right PIDs
6. PIDs analysis
7. Using the Software Heritage archive
8. Open Science
9. **Building for the long term**
10. Conclusion

Growing Support

Landmark Inria Unesco agreement, April 3rd, 2017

Inria
INVENTEURS DU MONDE NUMÉRIQUE

UNESCO

Sharing the vision

IEEE Software
GitLab
ENGINEERING
eclipse
OW2
fsfe
nlnet FOUNDATION
SIGSOFT
Liberté • Égalité • Fraternité RÉPUBLIQUE FRANÇAISE
INAPI
THE LINUX FOUNDATION
ADULLACT
INFORMATICS EUROPE
Computer History Museum
software freedom conservancy
Software Freedom Law Center
AdaCore The GNAT Pro Company
gandi.net
TDF
FREE SOFTWARE FOUNDATION
creative commons
SIF Société Informatique de France
open source initiative
openinventionnetwork

Contributing to the mission

Inria
INVENTEURS DU MONDE NUMÉRIQUE

>= 100Ke/year	Microsoft intel SOCIETE GENERALE
>= 50Ke/year	HUAWEI Google
>= 25Ke/year	DANS Data Archiving and Networked Services NOKIA Bell Labs ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA
>= 10Ke/year	GitHub UQÀM FOSSID

The next steps

The Software Heritage Foundation

- independent
- long term mission
- multistakeholder

The community

- academia: Open Access, research
- industry: better software
- cultural heritage: all the software history

The mirror network

- resilience
- biodiversity

> *"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."*
>
> *Thomas Jefferson*

You can help!

Connect infrastructures

- SWH and Zenodo
- SWH mirror at CERN

Contribute

- help broaden archive coverage
- forge.softwareheritage.org

EU Copyright directive: ACT NOW to protect software development!

savecodeshare.eu saveyourinternet.eu

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

Outline

1. Introductions
2. Software is everywhere…
3. … and we are not taking care of it!
4. The Software Heritage initiative
5. Looking for the right PIDs
6. PIDs analysis
7. Using the Software Heritage archive
8. Open Science
9. Building for the long term
10. Conclusion

Come in, we're open!

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