

Software Source Code Identification

Working Group

Roberto Di Cosmo

`roberto@dicosmo.org`

April 25nd, 2019



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

- 
- 1 Introduction
 - 2 Setting the stage
 - 3 Identifying software source code: motivations and difficulties
 - 4 Conceptual framework for identifiers
 - 5 IDOs : SWH-IDs
 - 6 DIOs and curation process: ASCL
 - 7 DIOs and curation via publications
 - 8 DIOs and IDOs: the HAL / SWH use case
 - 9 Discussion and wrap up

Working group key facts

Joint RDA & FORCE11 WG which spawned from
RDA's Software Source Code IG & FORCE11's SCIWG

Co-chairs

- Roberto Di Cosmo
- Daniel Katz
- Martin Fenner

Objectives

- bring together people involved/interested in software identification
- produce concrete recommendations for the academic community

Please register

online document: <http://bit.ly/rda13scidwg>

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

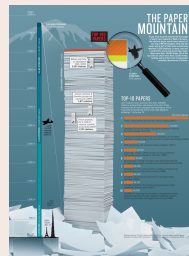
- 
- 1 Introduction
 - 2 Setting the stage
 - 3 Identifying software source code: motivations and difficulties
 - 4 Conceptual framework for identifiers
 - 5 IDOs : SWH-IDs
 - 6 DIOs and curation process: ASCL
 - 7 DIOs and curation via publications
 - 8 DIOs and IDOs: the HAL / SWH use case
 - 9 Discussion and wrap up

Software is *an essential component* of modern scientific research

Top 100 papers (Nature, October 2014)

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

<http://www.nature.com/news/the-top-100-papers-1.16224>



Software 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 Arena 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[1][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.”

An example from my research field, Computer Science

Repeatability in computer systems research, Christian Collberg, 2016

Analysis of 613 papers

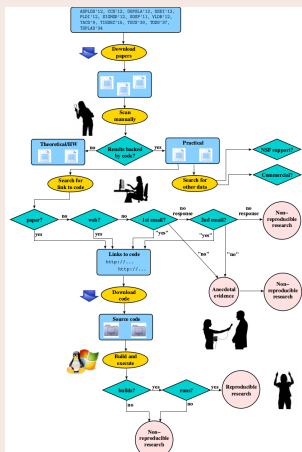
- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

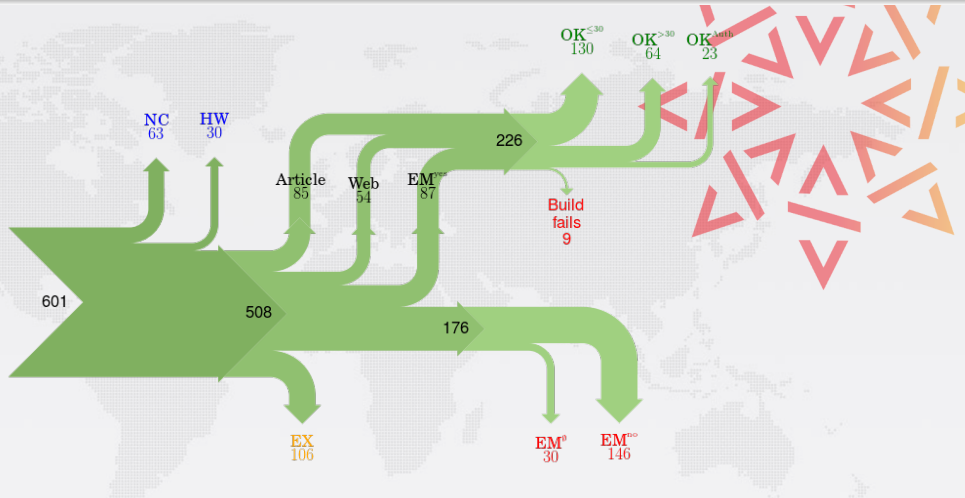
all very practical oriented

The basic question

can we get the code to build and run?

The workflow

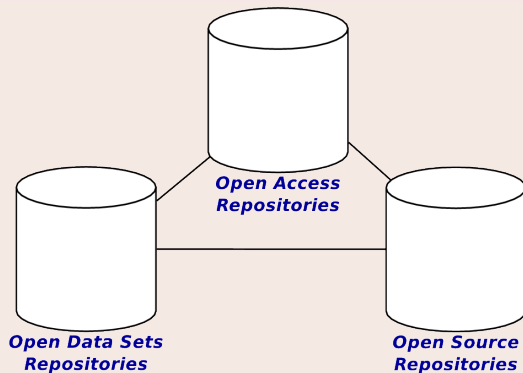




... that's a whopping 40% of **non reproducible** works!

Software Source code: important pillar of Open Science

The Magic Triangle of Scientific Knowledge



Nota bene

The links in the picture are **essential**

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/consensus on how to

- choose a license
- cite a software project
- relate to industry best practices
- make source code FAIR(*)

Lack of basic prerequisites to reproducibility

See a discussion in http://annex.softwareheritage.org/public/talks/2018/2018-09-17-STScI_public.pdf

Interest in (research) software is raising

A wealth of activities in academia

artifact evaluation

now commonplace in CS conferences

reproducible research

hot area of interest (jury still out on how to really do this)

software archival

publishers, open access portals, propose their services

academic credit

research software authors want recognition

Identifiers

for all the above, proper **identifiers** are needed

Challenges for academia

Accept the complexity: software is *special*

- made by humans for humans: copyright law applies!
- not (just) data: we may have a nice hammer, but software is not a nail

Industry, developers, communities have been there

we must

- avoid reinventing the wheel
- connect with existing communities of practice

Academic initiatives

- Force 11 Software Citation Implementation WG
- Freya EU project
- OpenAire EU project
- EU Open Science Monitor

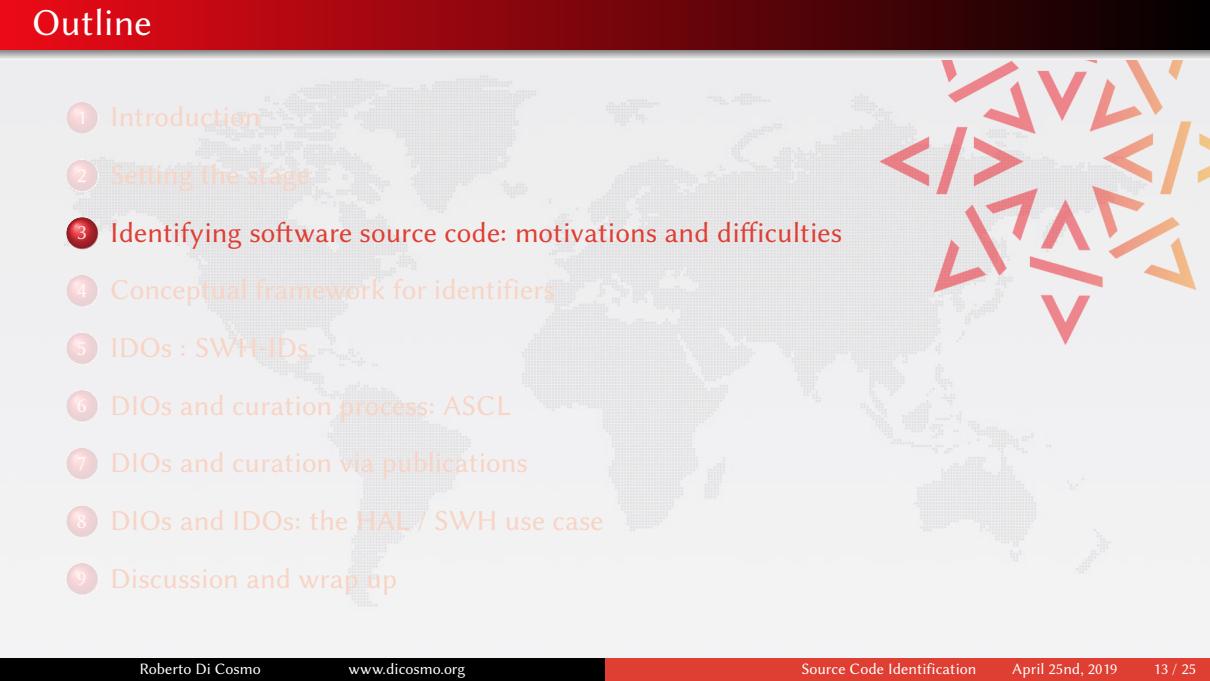
Industry initiatives

- NSRL (NIST)
- SPDX (Linux Foundation)
- SWID (ISO Standard)
- ...

Transversal initiatives

Software Heritage identifiers (SWH-ID)

(disclosure: I'm leading it)

- 
- 1 Introduction
 - 2 Setting the stage
 - 3 Identifying software source code: motivations and difficulties
 - 4 Conceptual framework for identifiers
 - 5 IDOs : SWH-IDs
 - 6 DIOs and curation process: ASCL
 - 7 DIOs and curation via publications
 - 8 DIOs and IDOs: the HAL / SWH use case
 - 9 Discussion and wrap up

Software source code identification

- motivations
- difficulties

The floor is yours

Support reproducible research and reuse

references to retrieve the exact version of a software artefact used in a research

Give credit

citations that count for software authors

Transparency

software bill of materials enable traceability of software artefacts

It is way more complex than it seems

All software projects are not born equal

structure

- monolithic
- composite

lifetime

- one shot
- long running

community

- single developer
- large community

authorship

- plurality of roles
- difficulty of evaluating contributions


authority

- just the commit log
- top down
- institution

And *attribution* adds to the complexity

software citation is much more than ...

software identification!

- 
- 1 Introduction
 - 2 Setting the stage
 - 3 Identifying software source code: motivations and difficulties
 - 4 Conceptual framework for identifiers**
 - 5 IDOs : SWH-IDs
 - 6 DIOs and curation process: ASCL
 - 7 DIOs and curation via publications
 - 8 DIOs and IDOs: the HAL / SWH use case
 - 9 Discussion and wrap up

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.)

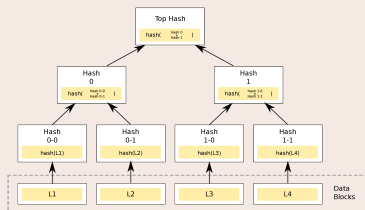
IDOs are enough for reproducibility

DIOs are needed for attribution

- 
- 
- 1 Introduction
 - 2 Setting the stage
 - 3 Identifying software source code: motivations and difficulties
 - 4 Conceptual framework for identifiers
 - 5 IDOs : SWH-IDs**
 - 6 DIOs and curation process: ASCL
 - 7 DIOs and curation via publications
 - 8 DIOs and IDOs: the HAL / SWH use case
 - 9 Discussion and wrap up

IDO in Software Development: the origins

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



Combination of

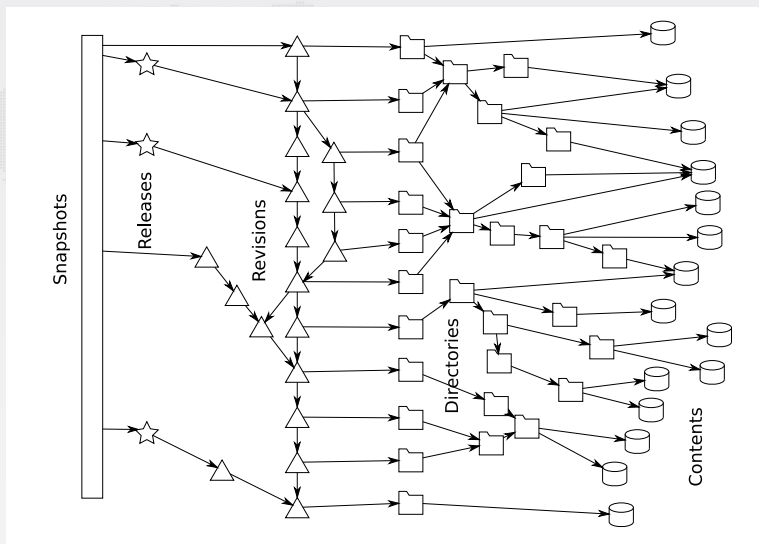
- tree
- hash function

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, ...)

IDO 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.

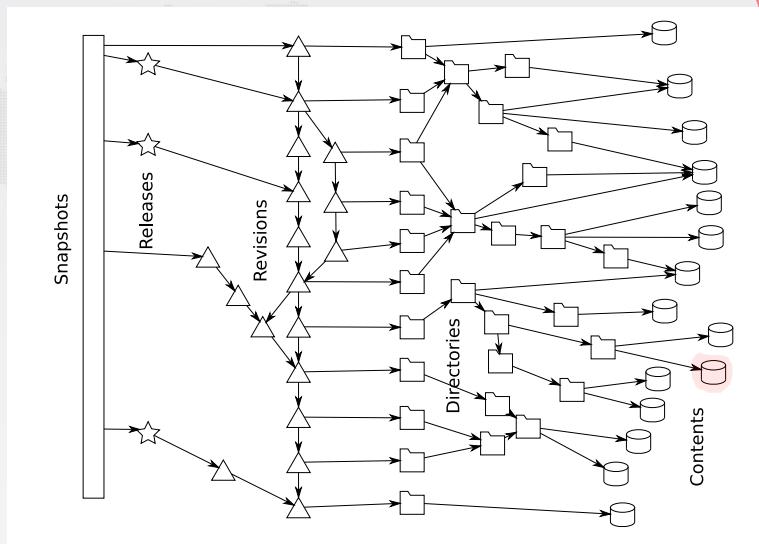
The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program—to make sure it remains free
software for all 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 things.

To protect your rights, we need to prevent

sha1: 8624bcdade55baeef...
sha256: 8ceb4b9ee5aded...
sha1_git: **94a9ed024d385...**
length: 35147

IDO in Software Heritage: a worked example



IDO in Software Heritage: a worked example

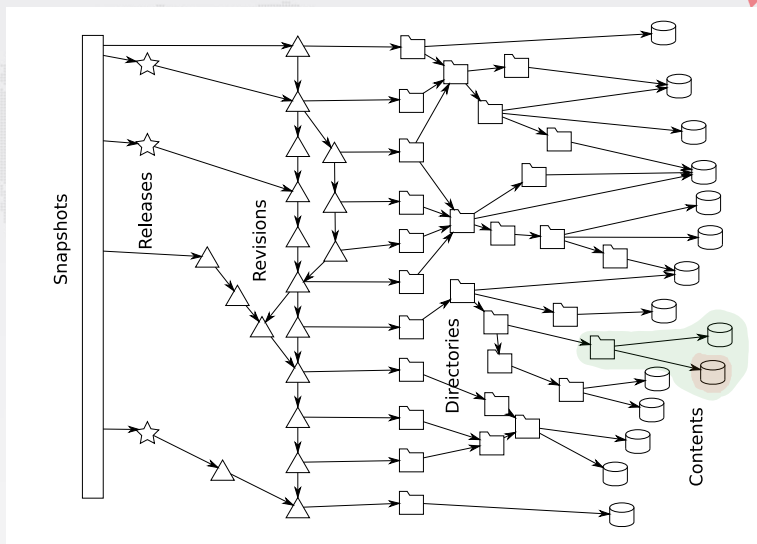


Directories


```
100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d0a34af6f52cf3cf6b0c2f7bd0648fbd255e77f AUTHORS
100644 blob 94a9ed024d3859793618152ea559a168bbcb5e2 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 eee147c36caf1bbcb2d820da8dc026cb5b68180bc setup.py
040000 tree 224bb4c1f4c67fca1d160bfdd2d06094e7e1abf3 sql
040000 tree 8631c9cd77bbe993168107ab5baf51f40c6300be swl
040000 tree 8fb905b56ba8ed692f1209b2773b474c6c1d66c1 utils
```

id: 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d

IDO in Software Heritage: a worked example



Revisions

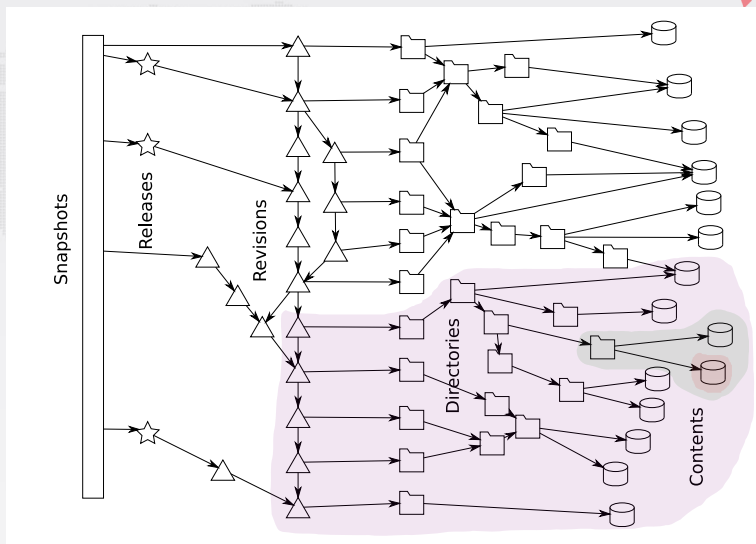
Details	Changes	Files
SHA: 963634dca6ba5dc37e3ee426ba091092c267f9f6		
Author: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Committer: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Subject: provenance.tasks: add the revision -> origin cache task		
Parent: 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

IDO 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

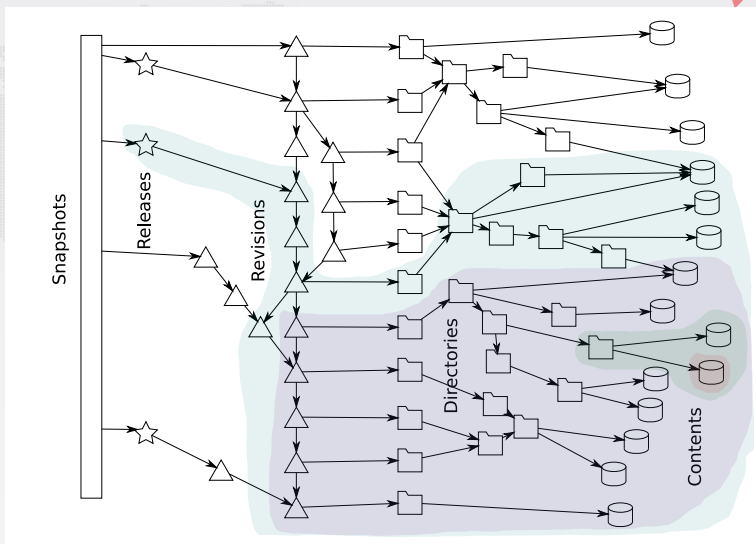
- Add new metadata column to origin_visit
- Update swh-add-directory script for updated API

-----BEGIN PGP SIGNATURE-----

iQIzBAABCAAdBQJXvZTNFhxuaWNvbGFzQGRhbmRyaW1vbnQuZXUACgkQ7AWLMo2+
neqonw/aq6SOB5DijzEa+kWN3rXgVS+1K1vEVh1wNKAwx8eKJ7aX2kEILDtt7uf
ahpZ6pz3q8nqs6aC1+YrxBfcih3L2YtrdZeWXWqr8xWNMaEoYDb8qaphwh8AD5t2
ICBlit2ujtXuCrDt93eKKPwvzZXg+h80sMWy3SDr6jW7Z7K4Mu/PgGlyLHPY55yo
IGEndWno7VFH1Vm6t1n5qB7I5mXRaqA+becqddubTZ2xij+jpUqC8cyqN3hm/fL
qsJ2mu8kyz3t8tG/H1/pV+I5OwBlNpOSSTH0tuojoEVgPK/dHSP79QuHDH2FkCao
klj6kAWyU80Mxb+nKV/jelBrR3+yWBFj3Qp5a1/V8oOT6E1dALcNMpEaKCoKtMt
d/gMRax1I1/g0EDfnsW67G6sDwKPKPHngfVLQ3nV3GaQQTnu1RpMz006H9/tAwzC
Gg/K1PdHT4hzOjI46wYPZyje0U2VXGFu6vVU9vFQ4ZR/Wjn+0zmZdcRdrJJSUOMN
RpTTfUusbXUeXHGOpkgXhSYTnvp1gdPc76U5TsK0aGe84AZm1Ik0mGrwXCVfPqIYo
nhhibB5HBNMoqyF6yTSOpUbYK70tpYRRUGKWDeRK0wKSxkWUzGtKzy6jYqJjo29
gulwqZqif5qWQCB0OntAL2+HvPFaVyckMejUhg62cP/+EHlvUk=
=kOxP
-----END PGP SIGNATURE-----

id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb

IDO in Software Heritage: a worked example



git show-refs

Snapshots

```
commit 08ffeb25770109525eb3ce21691466c53a1d9158 refs/heads/atime
commit ba5443a24e3f9fe323a46c292cec4fcbe61c67eb refs/heads/directory-listing-arrays
commit d69e0dbf892383ff6589b27fbc1c05d27238d9c5 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 586fba4e580b4f5fab05f599367643cbbc1a9c7f refs/tags/v0.0.15
tag 8cd8b885f4098bf363177742bd289f660e5be51c refs/tags/v0.0.16
tag a542444ee3f0fbcd35efb202fee035c809abc7d6 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 3fb168c2072a5d625124257a1e5dfc0f5ffa1df 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

- 
- 1 Introduction
 - 2 Setting the stage
 - 3 Identifying software source code: motivations and difficulties
 - 4 Conceptual framework for identifiers
 - 5 IDOs : SWH-IDs
 - 6 DIOs and curation process: ASCL**
 - 7 DIOs and curation via publications
 - 8 DIOs and IDOs: the HAL / SWH use case
 - 9 Discussion and wrap up

- 
- 1 Introduction
 - 2 Setting the stage
 - 3 Identifying software source code: motivations and difficulties
 - 4 Conceptual framework for identifiers
 - 5 IDOs : SWH-IDs
 - 6 DIOs and curation process: ASCL
 - 7 DIOs and curation via publications**
 - 8 DIOs and IDOs: the HAL / SWH use case
 - 9 Discussion and wrap up

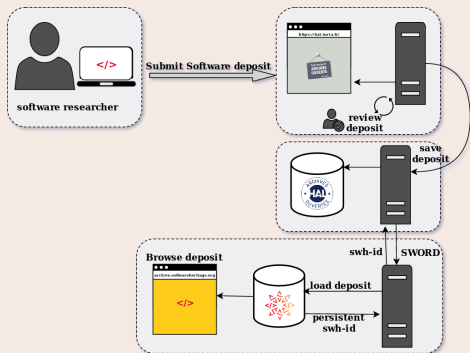
The swMath project

- see <https://swmath.org>
- nice example: <http://swmath.org/software/7116>
- source code is archived in Software Heritage

- 
- 1 Introduction
 - 2 Setting the stage
 - 3 Identifying software source code: motivations and difficulties
 - 4 Conceptual framework for identifiers
 - 5 IDOs : SWH-IDs
 - 6 DIOs and curation process: ASCL
 - 7 DIOs and curation via publications
 - 8 DIOs and IDOs: the HAL / SWH use case**
 - 9 Discussion and wrap up

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/>

- 
- 1 Introduction
 - 2 Setting the stage
 - 3 Identifying software source code: motivations and difficulties
 - 4 Conceptual framework for identifiers
 - 5 IDOs : SWH-IDs
 - 6 DIOs and curation process: ASCL
 - 7 DIOs and curation via publications
 - 8 DIOs and IDOs: the HAL / SWH use case
 - 9 Discussion and wrap up

Duration: 18 months

- collect state of the art
- extract minimum viable recommendations
- propose actionable plans

Curation process

talk to

- tech transfer departments
- promotion committees
- department reports

Identifiers

- thematic communities
-