Software Heritage at ZB MED colloquium

Archiving and Referencing all the source code towards recognizing software in academia

Morane Gruenpeter

Software engineer and metadata specialist
Inria, Software Heritage

morane@softwareheritage.org

December 10th, 2020

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

Short Bio: Morane Gruenpeter

Goal: Building the Semantic Web of Free and Open Source Software

2008-2011 B.A in Musique (Harpist)

2012-2015 Licence (B.SC) in Computer Science @CNAM

2015-2017 Master in Software Engineering (R&D) @UPMC

2017 Internship *Software Heritage* (SWH)

2018-2019 European project EU2020 *CROSSMINER* (on SWH team)

2020-2022 European project *FAIRsFAIR* (on SWH team)

Working groups for Open Science and digital preservation

- the Research Data Alliance's Software Source Code Interest Group (SSC IG),
- the FORCE11's Software Citation Implementation Working Group (SCI WG),
- the joint RDA & FORCE11 Software Identification Working Group (SCID WG)
- WikiData for Digital Preservation initiative (WikiDigi).

Living in Paris

Software is all around us

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

The World Wide Web, 1989, at CERN on a NeXT machine

“When somebody has learned how to program a computer … You’re joining a group of people who can do incredible things. They can make the computer do anything they can imagine.”

From An Insight, An Idea with Tim Berners-Lee (2013)

What is software ?

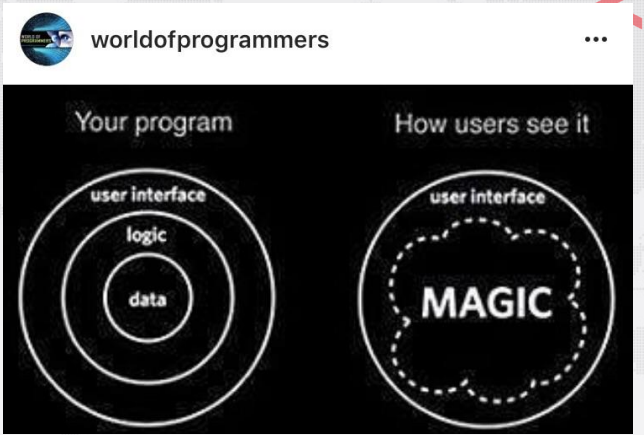


Image taken from from reddit - ProgrammerHumor

Software definition

Encyclopædia Britannica

"Software, instructions that tell a computer what to do. Software comprises the entire set of programs, procedures, and routines associated with the operation of a computer system. The term was coined to differentiate these instructions from hardware—i.e., the physical components of a computer system." link

Software as a concept

- software project / entity
- the creators and the community around it
- the software idea / algorithms / solutions

Software artifact

- the executable (or binary) of each version for a specific environment
- the software source code for each revision

Much more complex than it seems

Software is complex

Structure	monolithic/composite; self-contained/external dependencies
Lifetime	one-shot/long term
Community	one man/one team/distributed community
Authorship	complex set of roles
Authority	institutions/organizations/communities/single person

Various granularities

Exact status of the source code for reproducibility, e.g.

"you can find at swh:1:cnt:cdf19c4487c43c76f3612557d4dc61f9131790a4;lines=146-187 the core algorithm used in this article"

(Major) release *"This functionality is available in OCaml version 4"*

Project *"Inria has created OCaml and Scikit-Learn"*.

Is this *software*?

Ceci n'est pas une pipe.

What about *software source code* ?

Outline

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");
}
```

Source code is *special*

Executable and *human readable* knowledge — copyright law

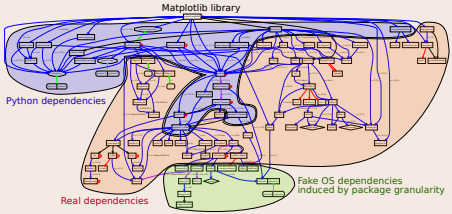
"Programs must be written for people to read, and only incidentally for machines to execute."
Harold Abelson

Software *evolves* over time

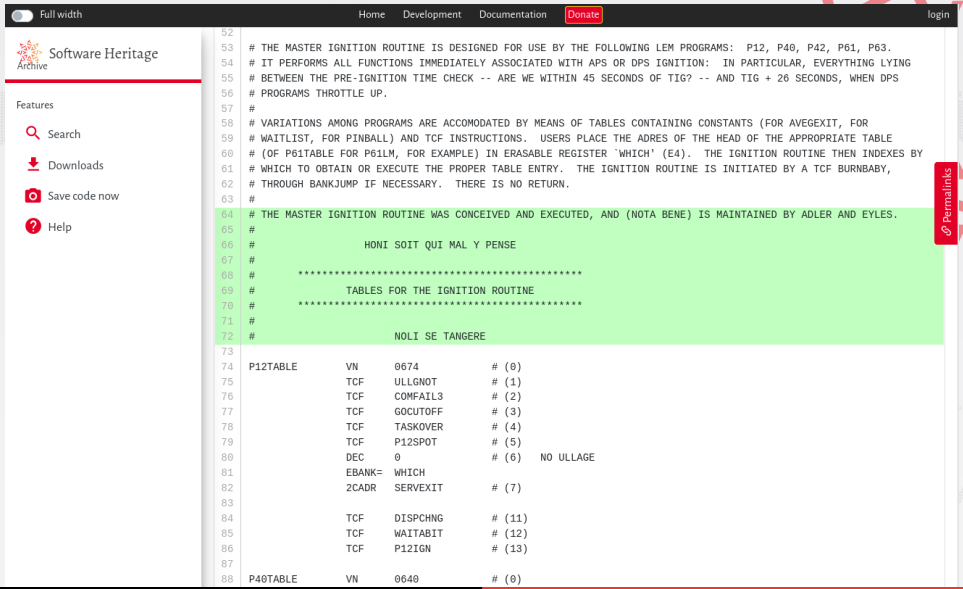
- projects may last decades
- the *development history* is key to its *understanding*

Complexity

- *millions* of lines of code
- large *web of dependencies*
  - easy to break, difficult to maintain
- sophisticated *developer communities*



Software Source Code human readable and executable knowledge

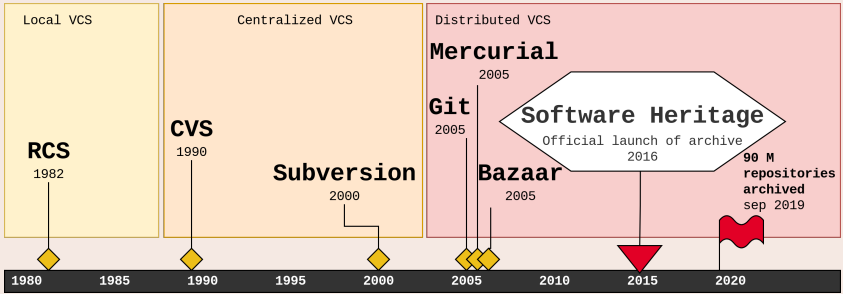


Version Control System timeline

Version control system (VCS)

- records changes made to a (set of) *source code file* (s)
- allows to operate on versions: diff/merge/fork/recover etc.
- essential tool for software development

Three decades of evolution



Outline

Software Heritage in a nutshell

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Collect, preserve and share *all* software source code

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

Reference catalog

find and reference all software source code

Universal archive

preserve all software source code

Research infrastructure

enable analysis of all software source code

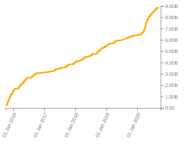
Cultural Heritage | Industry | Research | Education



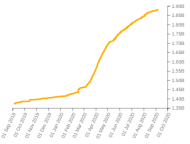

Software Heritage

As of today the archive already contains and keeps safe for you the following amount of objects:

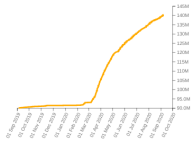
Source files
8,846,381,610



Commits
1,880,663,008



Projects
140,348,311



Directories
7,506,954,410

Authors
38,603,337

Releases
15,051,940

Growing Support

Raising awareness: landmark agreement, 3/4/2017; grand opening, 7/6/2018

Inria — INVENTEURS DU MONDE NUMÉRIQUE

UNESCO

Sharing the vision

The Software Heritage, 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, OPEN SOURCE, openinventionnetwork

Sponsoring our work

Inria — INVENTEURS DU MONDE NUMÉRIQUE

Platinum sponsors

Microsoft, intel, SOCIETE GENERALE, HUAWEI

Gold sponsors

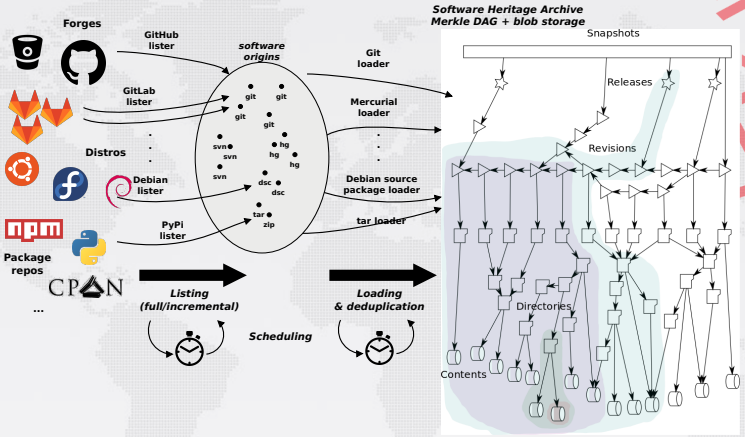
openinventionnetwork, Université de Paris, SORBONNE UNIVERSITÉ

Silver sponsors

GitHub, Google, Università di Pisa, vmware

Outline

Under the hood: Automation, and storage

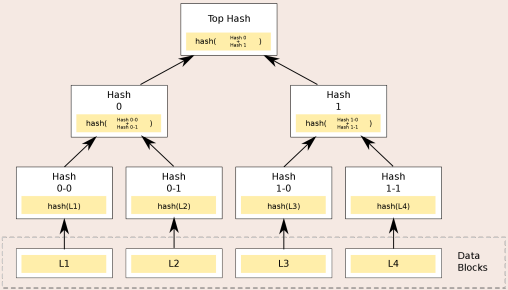


Global development history permanently archived in a uniform data model

- over 6 billion unique source files from over 90 million software projects
- ~400 TB (uncompressed) blobs, ~20 B nodes, ~280 B edges

Much more than an archive!

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



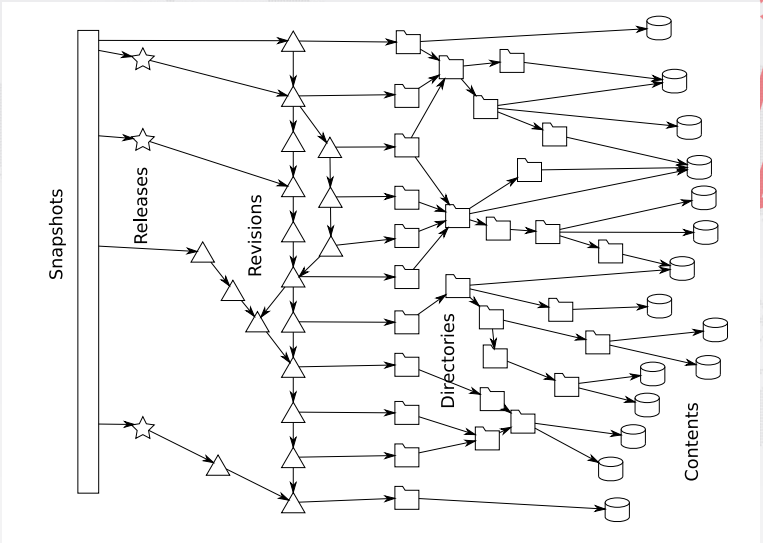
Combination of

- tree
- hash function

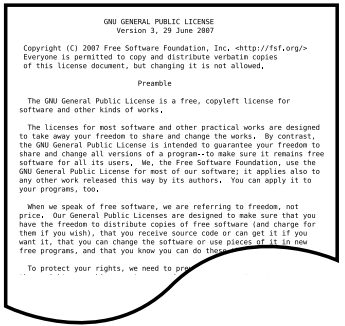
Classical cryptographic construction

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

The archive in pictures



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.

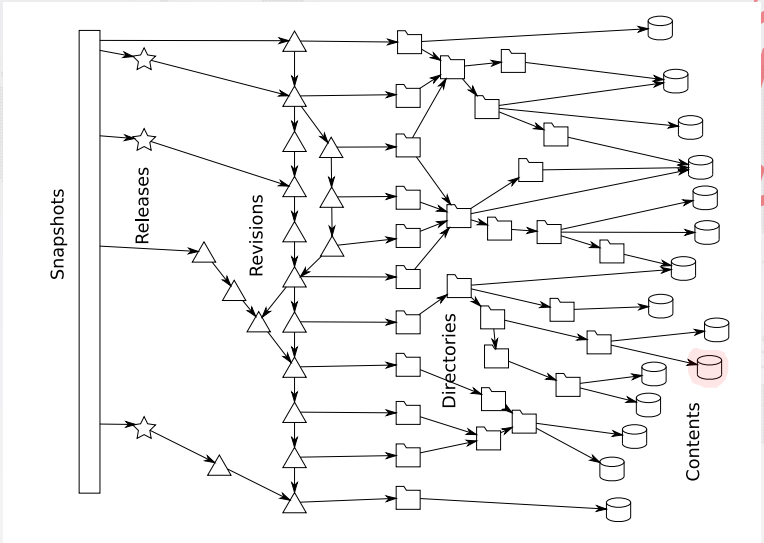
The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all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

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

The archive in pictures

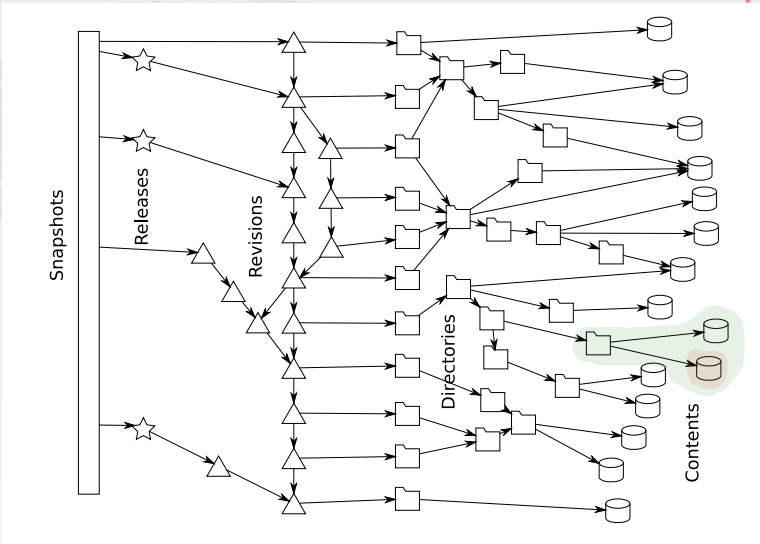


Directories

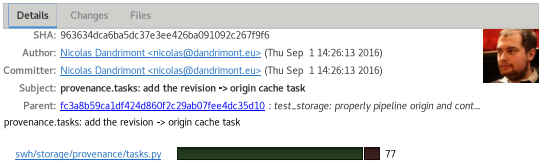
```
100644 blob c5baade4c44766042186ef858c0fd63d587ebf09 .gitignore
100644 blob 2d0a34af6f52cf3cf6b0c2f7bd0648fbd255e77f AUTHORS
100644 blob 94a9ed024d3859793618152ea559a168bbcbb5e2 LICENSE
100644 blob d9b2665a435a43f8a79a84e0867751dfb095c7bb MANIFEST.in
100644 blob 524175c2bad0b35b975f79284c2f5a6d5eaf2eb4 Makefile
100644 blob 5c7e3a5bbddb038682ba7793f440492ed9678bb3 Makefile.local
100644 blob 8617980629cd24e6080404f09aa749b085b3e07b README.db_testing
100644 blob 76b29f94cf815e0869c414d38d78d7ce08ec514e README.dev
040000 tree 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

The archive in pictures



Revisions

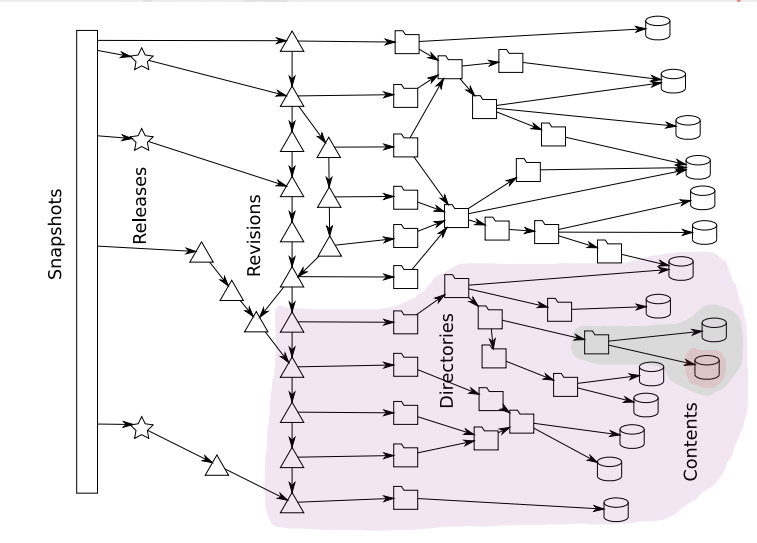


tree 515f00d44e92c65322aaa9bf3fa097c00ddb9c7d
parent fc3a8b59ca1df424d860f2c29ab07fee4dc35d10
author Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200
committer Nicolas Dandrimont <nicolas@dandrimont.eu> 1472732773 +0200

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

id: 963634dca6ba5dc37e3ee426ba091092c267f9f6

The archive in pictures



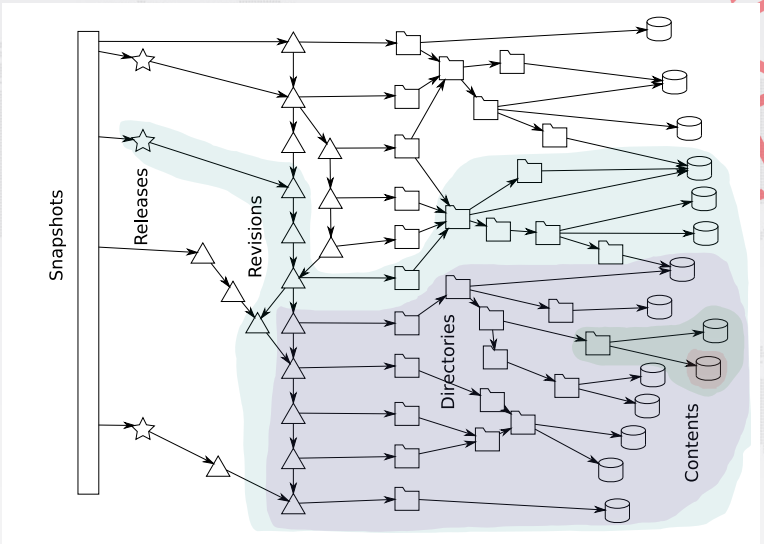
Releases

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

Release 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
ICBIlt2ujtXuCrDt93eKKPwvzZXq+hB0sMWy35Dr6jW7Z7K4Mu/PGglyIHPY55yo
IGEndWno7Vfh1Vm6t1n5qB7I5mXRaqA+becqddubTZ2xjj+jpIUqC8cyqN3hm/fL
qsj2mu8kyz3t8tG/H1/pV+I5OwBlnPoS5TH0tujoJEVgPK/dHSP79QuHDHZFkCao
klj6kAWyU80Mxb+nKV/jeLbrR3+yWBFj3Qp5a1/V8oOTh6E1dALcNMpEaKCoKtMt
d/gMRax1I1/g0EDfnsW67G6sDwKPKPHhgfVLQ3nV3GaQQTnu1RpMzO06H9/AwzC
Gq/K1PdHT4hzOIl46wYPZyje0U2VXGFu6vVU9vFQ4ZR/WJn+0zMzdcRdRJ5UOMn
RpTTfUsbXUeXHGOpkgXhSYTnvp1gdPc76USTsK0aGe84AZm1Ik0mGrwXCVfPqlYo
nhhibBSHBNMoqyF6yTSOpUbYK70tpYRRUGKWDeRK0wKSxkWkUZGtKzy6jYqJjo29
gulwgZQif5qWQCB0OontAL2+HvPFaVyckMejUhg62cP/+EHIvUk=
=kOxP
-----END PGP SIGNATURE-----
```

id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb

The archive in pictures



git show-refs

Snapshots

```
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 3590e0ca0eb070e5b376705fa230bbfa4ffa5cc refs/tags/v0.0.11
tag 33378427a403ba569a67777b8d58f667dfbc6556 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 8cd8bb885f4098bf363177742bd289f660e5be51c 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

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*)

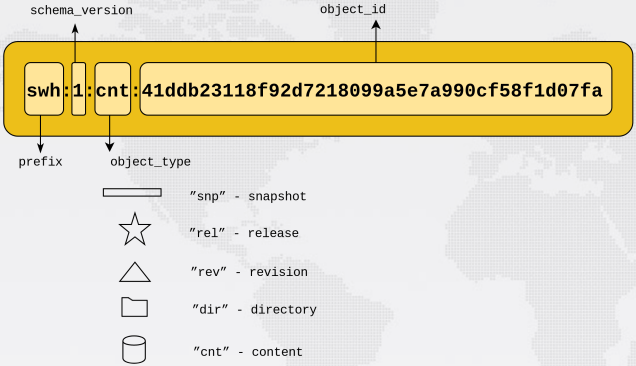
no middle man no central authority is needed (sw dev, *reproducibility*)

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

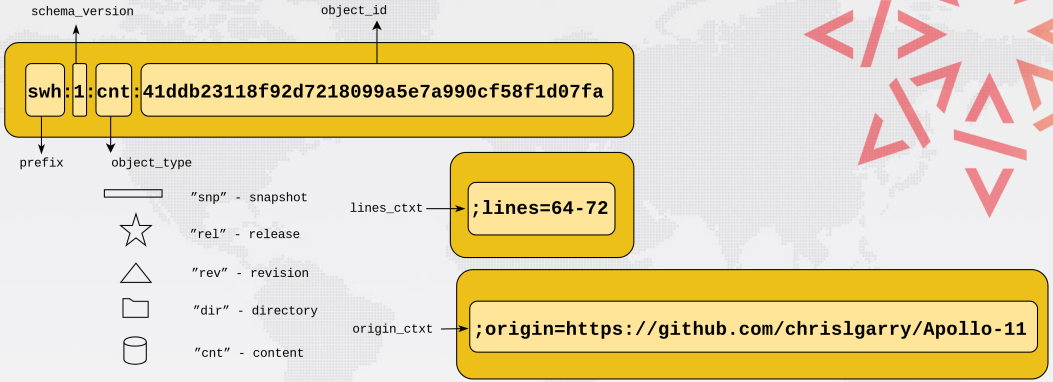
Intrinsic, decentralised, cryptographically strong identifiers = SWHIDs

The SWH-ID schema

The SWH-ID schema



The SWH-ID schema



Demo time

Let's look at some famous exceprts of source code

Apollo 11 source code (excerpt)

```
P63SPOT3        CA      BIT6            # IS THE LR ANTENNA IN POSITION 1 YET
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     P63SPOT4        # BRANCH IF ANTENNA ALREADY IN POSITION 1

                CAF     CODE500         # ASTRONAUT:    PLEASE CRANK THE
                TC      BANKCALL        #               SILLY THING AROUND
                CADR    GOPERF1
                TCF     GOTOPOOH        # TERMINATE
                TCF     P63SPOT3        # PROCEED       SEE IF HE'S LYING

P63SPOT4        TC      BANKCALL        # ENTER         INITIALIZE LANDING RADAR
                CADR    SETPOS1

                TC      POSTJUMP        # OFF TO SEE THE WIZARD ...
                CADR    BURNBABY
```

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;
}
```

It works!

we have *intrinsic* identifiers for all 20+ billion objects in the archive

Outline

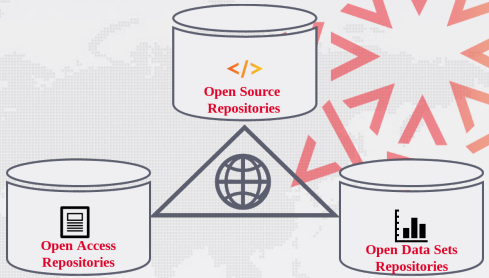
Software is a *forgotten* pillar of Open Science

Lack of recognition

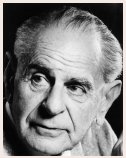
not (yet) a first class output

- in the EOSC plan
- in the scholarly world

Sometimes, if you don't have the software, you don't have the data
Christine Borgman, Paris, 2018



Reproducibility is the key



non-reproducible single occurrences are of no significance to science

Karl Popper, The Logic of Scientific Discovery, 1934

A plurality of needs

Researchers

- archive and reference software used and created in articles
- find useful software
- get credit for developed software
- verify/reproduce/improve results

Laboratories/teams

- track software contributions
- produce reports
- maintain web page

Research Organization

- know its software assets for technology transfer, impact metrics and strategy.

Software in research has different roles

Multiple facets, it can be seen as:

- a tool
- a research outcome or result
- the object of research

By identifying the software role

we can decide how to *treat* it

Archival

Research software artifacts must be properly archived

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

Identification

Research software artifacts must be properly referenced

make it sure we can *identify* them (*reproducibility*)

Metadata

Research software artifacts must be properly described

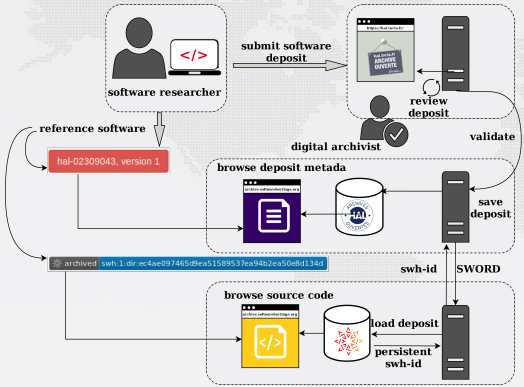
make it easy to *discover* them (*visibility*)

Citation

Research software artifacts must be properly cited *(not the same as referenced!)*

to give *credit* to authors (*evaluation*!)

The research software (deposit) use case



Deposit software in HAL — poster

Generic mechanism:

- SWORD based
- review process
- versioning

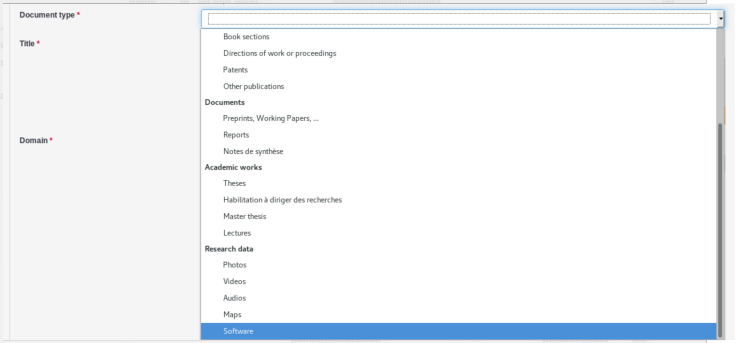
How to do it: (*guide*)

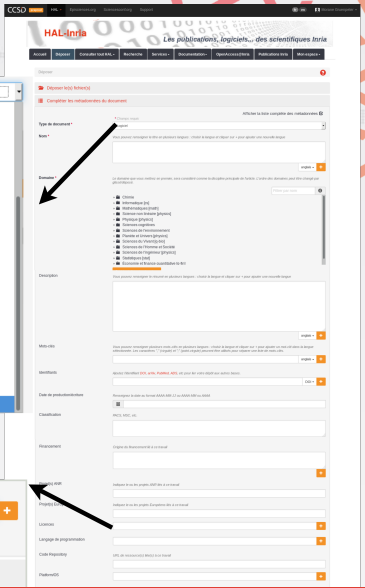
- deposit .zip or .tar.gz file with metadata

Timeline:

- *March 2018*: test phase on HAL-Inria
- *September 2018*: open to all HAL
- *December 2019*:
  - 80 complete source code deposits
  - 98 software records

Submit your source code (*guide*)





The deposit view

Home | Submit | Browse all HAL | Inria publications | Search | Services | Help | OpenAccess

hal-02983420, version 1

CGAL 3D Periodic Mesh Generation

Mikhail Bogdanov [1] , Aymeric Pellé [1] , Mael Rouxel-Labbé [2] , Monique Teillaud [3] Details

1 GEOMETRICA - Geometric computing
CRISAM - Inria Sophia Antipolis - Méditerranée , Inria Saclay - Ile de France
2 GeometryFactory
3 GAMBLE - Geometric Algorithms and Models Beyond the Linear and Euclidean realm
Inria Nancy - Grand Est, LORIA - ALGO - Department of Algorithms, Computation, Image and Geometry

Abstract : This package is devoted to the generation of isotropic simplicial meshes discretizing periodic 3D domains. The domain to be meshed is a region of the three-dimensional flat torus with cubic fundamental domain.

Document type : Software
Domain : Computer Science [cs] / Computational Geometry [cs.CG]

Complete list of metadatas Display

BROWSE

Software Heritage

swh:1:dir:0a5e5b721c21330f31a0511621a934d9aec38e4f;origin=https://github.com/CGAL/cgal;visit=swh:1:snp:78e145aa8174e576786284475a76cf6f187b3475;anchor=swh:1:rev:b86a5018c7f5f733c80fe40eee65803c112f2685;path=/Periodic_3_mesh_3/

Browse

https://hal.inria.fr/hal-02983420
Contributor : Monique Teillaud <Monique.Teillaud@inria.fr>
Submitted on : Thursday, October 29, 2020 - 5:57:43 PM
Last modification on : Sunday, November 1, 2020 - 3:24:41 AM

METADATA

version CGAL 4.13

Software License GNU General Public License v3.0 or later

Programming Language C++

Code Repository https://github.com/CGAL/cgal/tree/master/Periodic_3_mesh_3

Platform/OS Multiplatform

COLLECTIONS

UNIV-LORRAINE | LORIA | LORIA-ALGO | CNRS | INRIA | UNIV-PARIS-SACLAY

CITATION

Mikhail Bogdanov, Aymeric Pellé, Mael Rouxel-Labbé, Monique Teillaud. CGAL 3D Periodic Mesh Generation. 2018, (swh:1:dir:0a5e5b721c21330f31a0511621a934d9aec38e4f;origin=https://github.com/CGAL/cgal;visit=swh:1:snp:78e145aa8174e576786284475a76cf6f187b3475;anchor=swh:1:rev:b86a5018c7f5f733c80fe40eee65803c112f2685;path=/Periodic_3_mesh_3/). (hal-02983420)

SHARE

Reference vs. citation

Credit & Attribution

- a metadata record
- all authors & contributors

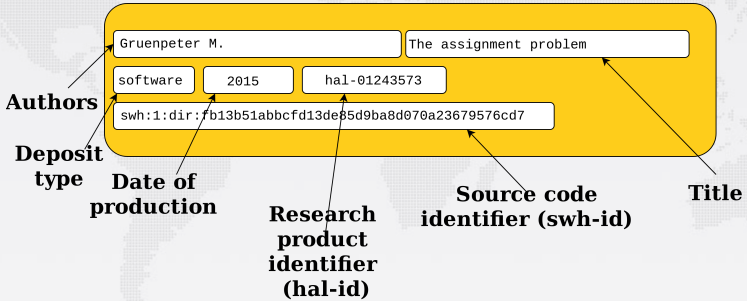
Reuse & Reproducibility

- a specific artifact
- with complementary information (docs)

Archive & Index

- metadata record (HAL)
- artifact itself (SWH)

connect the dots…



Prepare your public repository with:

- README, LICENSE, AUTHORS & codemeta.json files

What's a good README

extracted from Eric Steven Raymond and Make a README

MUST include:

- Name and a description of the software.

SHOULD include:

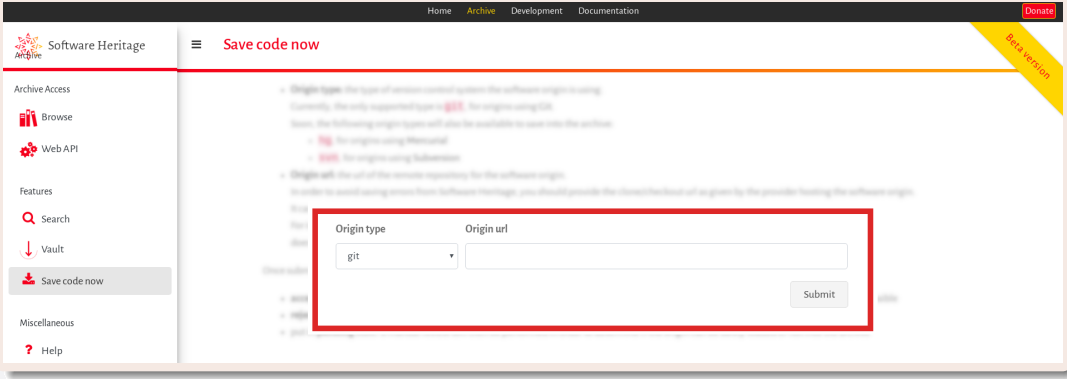
- how to run and use the source code
- build environment, installation, requirements

CAN include:

- project website or documentation pointer and recent news
- visuals

Save code now on https://archive.softwareheritage.org/save/

- git, svn or mercurial
- intrinsic metadata files
- complete history



Choose the granularity level for the reference:

file (with code fragment)

swh:1:**cnt**:c60366bc03936eede6509b23307321faf1035e23;lines=473-537
... and add *;origin=https://github.com/sagemath/sage/*

James McCaffrey's **algorithm** in sageMath

directory

swh:1:**dir**:c6f07c2173a458d098de45d4c459a8f1916d900f
... and add *;origin=https://github.com/id-Software/Quake-III-Arena/*

source code of **Quake-III Arena** from id-Software

specific release

swh:1:**rel**:22ece559cc7cc2364edc5e5593d63ae8bd229f9f
... and add *;origin=https://github.com/darktable-org/darktable/*

release 2.3.0 of Darktable, dated 24 December 2016

full snapshot (including all branches and all releases)

swh:1:**snp**:c7c108084bc0bf3d81436bf980b46e98bd338453
... and add *;origin=https://github.com/darktable-org/darktable/*

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

Outline

The metadata challenge - Software ontologies

"Ontologies are agreements, made in a social context, to accomplish some objectives. It's important to understand those objectives, and be guided by them.\"

T. Gruber, The Pragmatics of Ontology, 2003

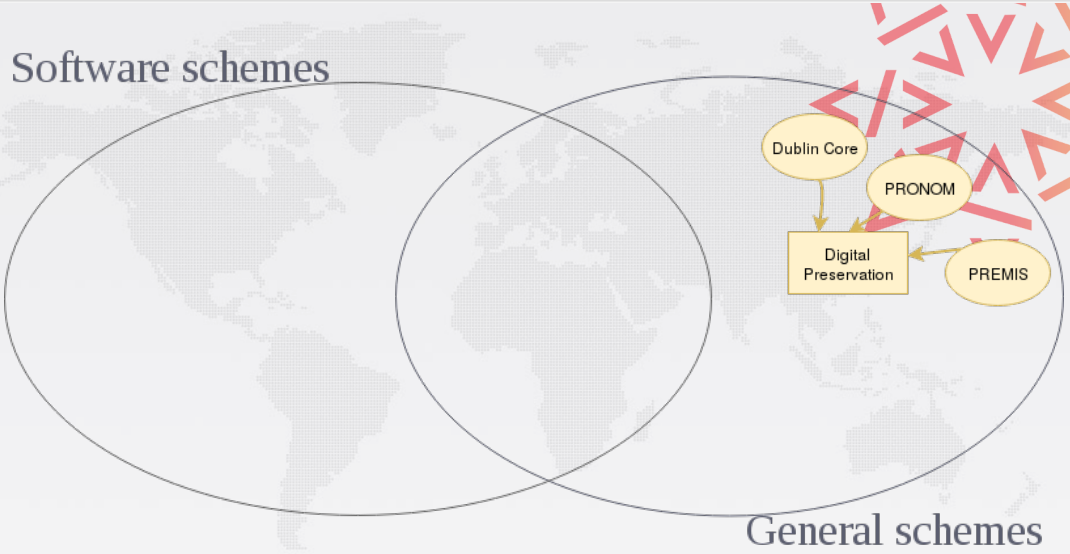
What do we want to describe?

- a software project?
- a software artifact? a collection of artifacts?
- With what terms or vocabulary?

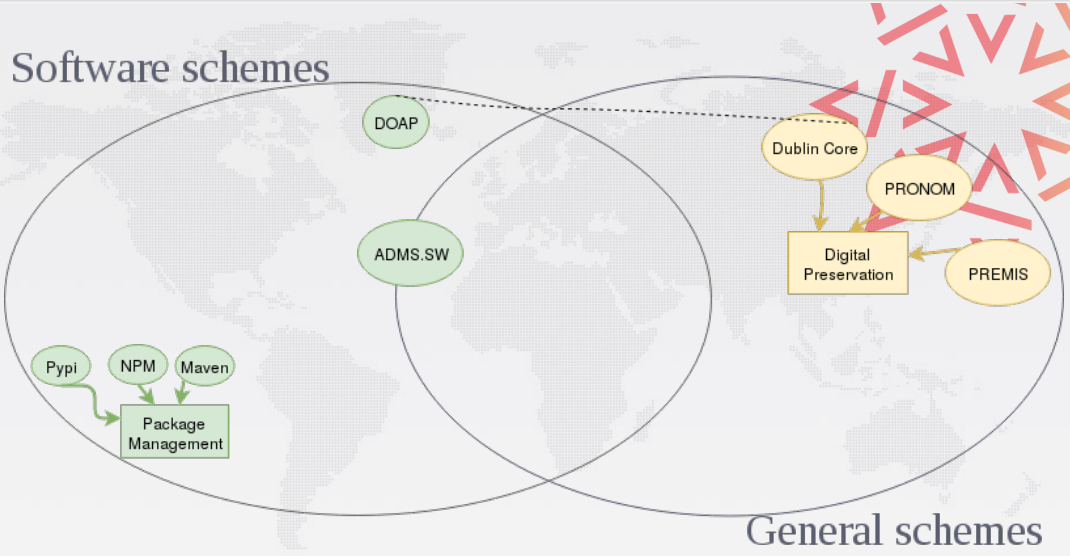
Software Citation Principles (Smith et al. 2016)

- Importance : first class citizen in the scholarly ecosystem
- Credit and attribution : authors, maintainer
- Unique identification: points to a unique, specific software version (DOI, Git SHA1 hash, etc..)
- Persistence : identification beyond the lifespan of the software (swh-id)
- Accessibility: url, publisher
- Specificity : version, environment

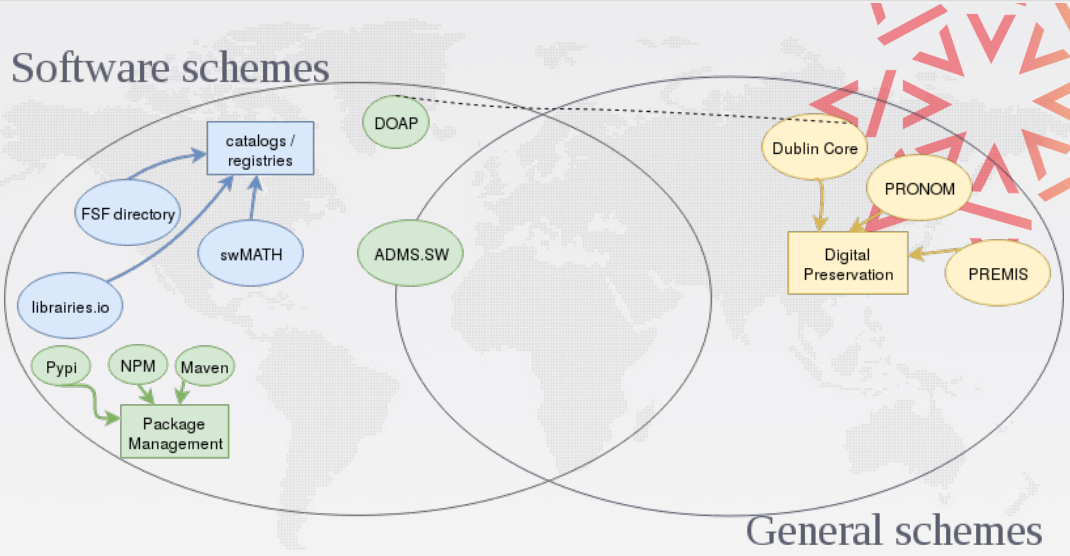
The metadata landscape



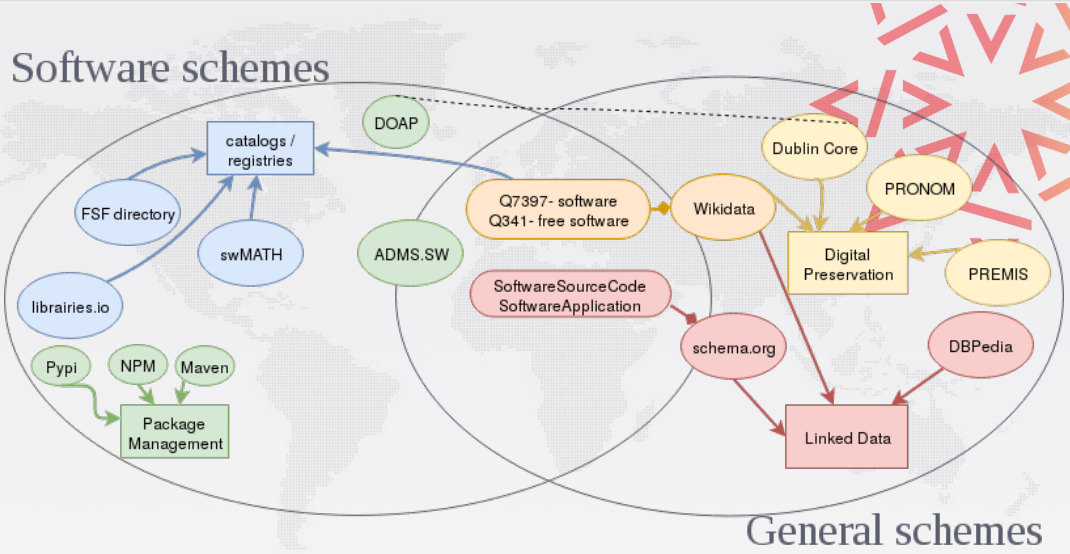
The metadata landscape



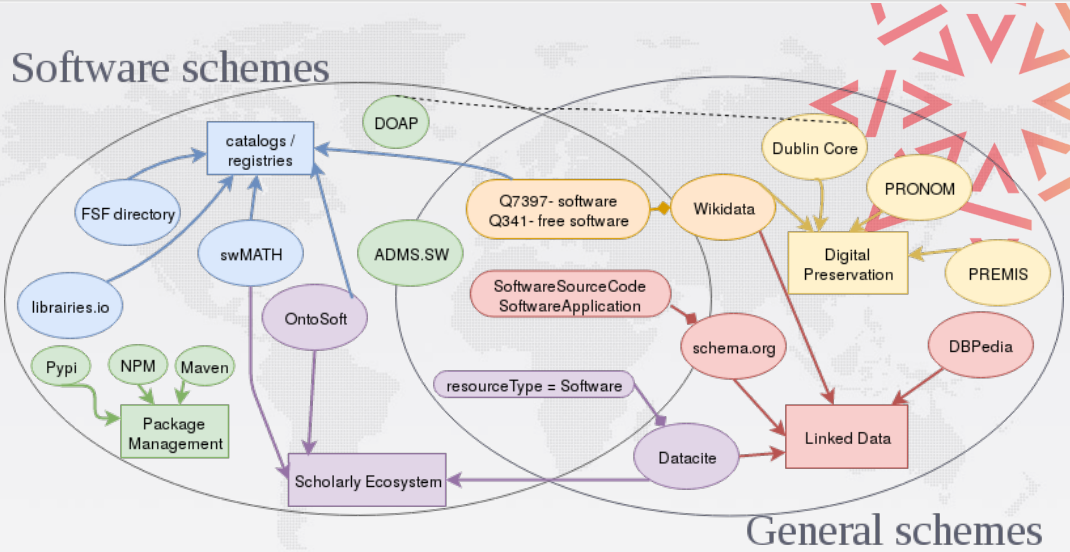
The metadata landscape



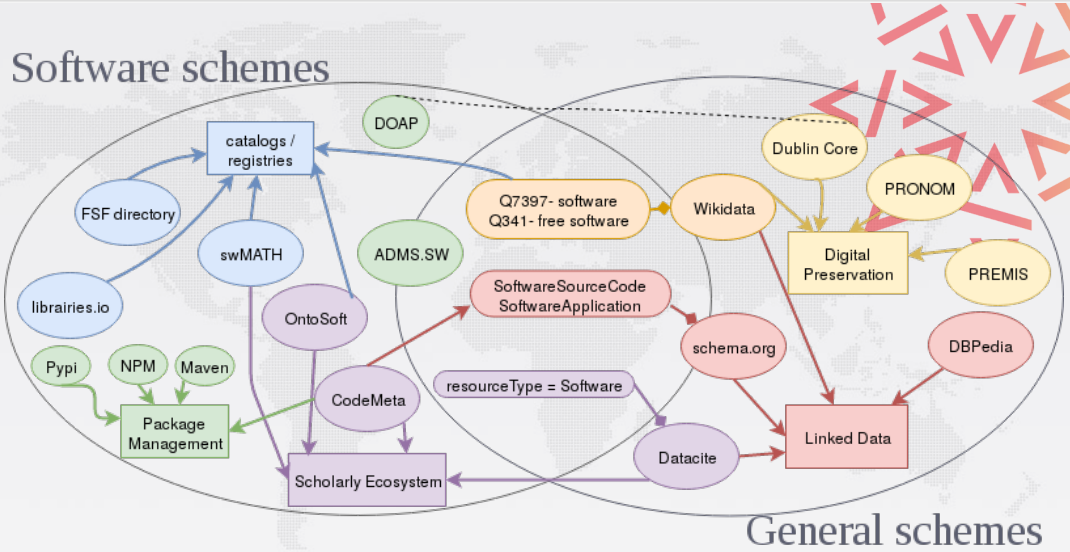
The metadata landscape



The metadata landscape



The metadata landscape



identify

- identifier
- name
- author(s)
- contributor(s)
- version
- applicationCategory
- codeRepository

administrate

- maintainer (contact*)
- citation
- funder(s)
- license
- editor / publisher
- dates (created, modified, published)
- developmentStatus

execute

- buildInstructions
- issueTracker
- operatingSystem
- softwareRequirements
- runtimePlatform
- downloadUrl
- (memory, procesor, storage)

classify

- description
- keywords
- supportingData
- referencePublication
- algorithms*
- readme (docs*)

Most fields are optional. Mandatory fields will be highlighted when generating Codemeta.

The software itself

Name

My Software

the software title

Description

My Software computes ephemerides and orbit propagation. It has been developed from early ´80.

Creation date

YYYY-MM-DD

Outline

Come in, we're open!

Software Heritage

- universal source code archive
- intrinsic identifiers (SWHIDs)
- open, non profit, long term
- infrastructure for Open Science

You can help improve science!

- use SWH and save *relevant* source code
- build on SWH (see swmath.org and ipol.im)
- contribute to SWH- it is *open source*
- spread the word

Software Heritage

Thank you! Any questions?

contact: morane@softwareheritage.org

P. Alliez, R. Di Cosmo, B. Guedj, A. Girault, M. Hacid, A. Legrand, N. Rougier
Attributing and Referencing (Research) Software: Best Practices and Outlook From Inria
Computing in Science & Engineering, 22 (1), pp. 39-52, 2020, ISSN: 1558-366X

Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli
Referencing Source Code Artifacts: a Separate Concern in Software Citation
Computing in Science & Engineering, 2020, ISSN: 1521-9615