

Software Heritage

Building the Software Pillar of Open Science

Stefano Zacchioli

4 Nov 2025

Open Access Week, Institut Polytechnique de Paris
Palaiseau, France



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

- 1 Introduction
- 2 (Open) Source for Open Science
- 3 Software Heritage in the big picture
- 4 Demo time
- 5 Much more than an archive
- 6 Conclusion

- Professor of Computer Science, Télécom Paris, Polytechnic Institute of Paris
- Free/Open Source Software activist (20+ years)
- Debian Developer & Former 3x Debian Project Leader
- Former Open Source Initiative (OSI) director
- Software Heritage co-founder & chief scientific officer

Software Source Code is Precious Knowledge

Harold Abelson, Structure and Interpretation of Computer Programs (1st ed.)

1985

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

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      GOTOP00H      # 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
```

Covid Sim (excerpt)

```
/**
 * @brief The basic unit of the simulation and is associated to a geographical location.
 *
 * Interventions (e.g., school closures) are tracked at this level. It contains a list of its
 * members (people), places (schools, universities, workplaces etc.), road networks, links to
 * airports etc.
 */
struct Microcell
{
    /* Note use of short int here limits max run time to USHRT_MAX*ModelTimeStep - e.g. 65536*0.25=16384 days=44 yrs.
     * Global search and replace of 'unsigned short int' with 'int' would remove this limit, but use more memory.
     */

    int n; // Number of people in microcell
    int adunit; // admin unit microcell belongs to
    int* members; // array of members/hosts of microcell

    int* places[MAX_NUM_PLACE_TYPES]; // list of places (of various place types) within microcell
    unsigned short int NumPlacesByType[MAX_NUM_PLACE_TYPES]; // number of places (of various place types) within microcell
    unsigned short int keyworkerproph, move_trig, place_trig, socdist_trig, keyworkerproph_trig;
    unsigned short int move_start_time, move_end_time;
    unsigned short int place_end_time, socdist_end_time, keyworkerproph_end_time;
    TreatStat moverest, treat, vacc, socdist, placeclose;
    unsigned short int treat_trig, vacc_trig;
    unsigned short int treat_start_time, treat_end_time;
    unsigned short int vacc_start_time;
    IndexList* AirportList;
};
```

Len Shustek, Computer History Museum

2006

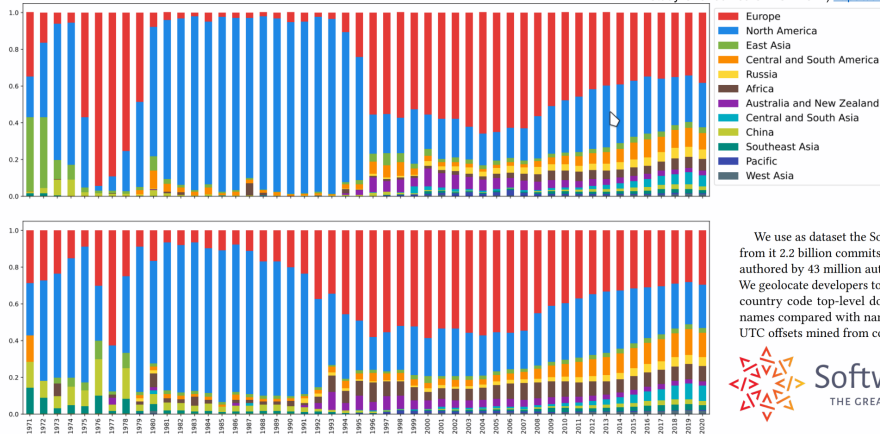
“Source code provides a view into the mind of the designer.”

(Open) Source Code comes from all over the world

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA

Davide Rossi and Stefano Zacchiroli

Geographic Diversity in Public Code Contributions: An Exploratory Large-Scale Study Over 50 Years. MSR 2022) <https://doi.org/10.1145/3524842.3528471>



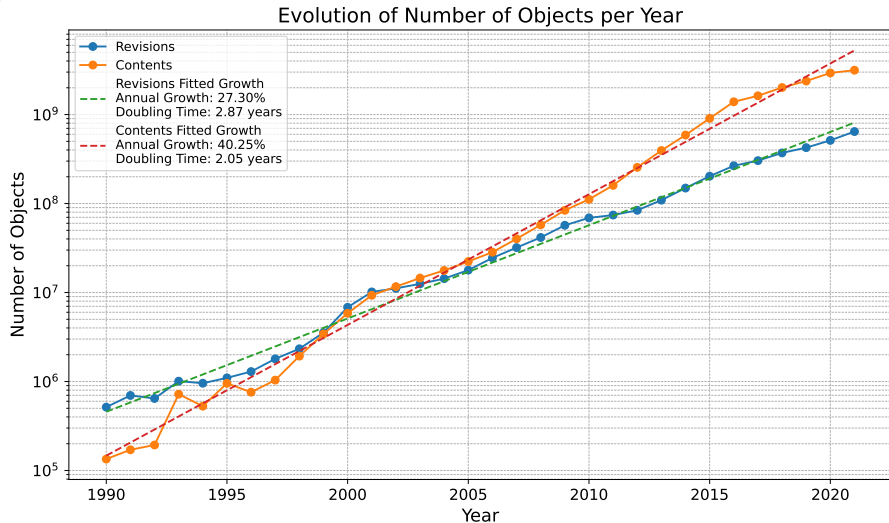
We use as dataset the Software Heritage archive [3] and analyze from it 2.2 billion commits archived from 160 million projects and authored by 43 million authors during the 1971–2021 time period. We geolocate developers to 12 world regions, using as signals email country code top-level domains (ccTLDs) and author (first/last) names compared with name distributions around the world, and UTC offsets mined from commit metadata.



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Figure 3: Ratio of commits (above) and active authors (below) by world zone over the 1971–2020 period.

(Open) Source Code grows at an exponential rate



Rousseau, Di Cosmo, Zacchioli: Software provenance tracking at the scale of public source code. *Empir. Softw. Eng.* 25(4): 2930-2959 (2020)

- 1 Introduction
- 2 (Open) Source for Open Science
- 3 Software Heritage in the big picture
- 4 Demo time
- 5 Much more than an archive
- 6 Conclusion

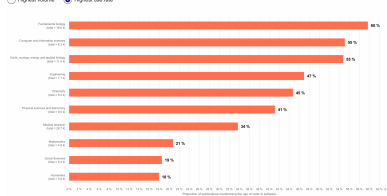
Software Source Code: pillar of Open Science

Software powers modern research

Proportion of publications in France that mention the use of code or software by discipline

Sort by:

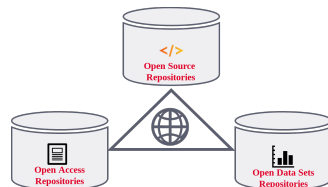
☐ Highest volume ☒ Highest use ratio



Over 20% of articles using software across all disciplines share it

2024 French Open Science Monitor

Key pillar: software



Links are **important**

Nota Bene

software may be a *tool*, a *research outcome* and a *research object*

access to the *source code* is essential!

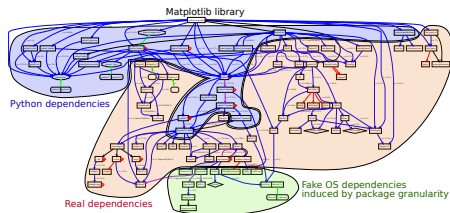
Source code is *special* (software is *not* data)

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
 - *research software* a thin top layer
- sophisticated *developer communities*



Precious, endangered *executable* and *human readable* knowledge

key people **passing away**, platforms (GoogleCode, Gitorious, etc.) closing down ...

no organised effort to catalog and archive it

Paris Call on Software Source code (2019, UNESCO)



40 international experts call to “promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, [...] recognising in the careers of academics their contributions to high quality software development, in all their forms”

UNESCO recommendations for Open Science, 2018-2021

“The source code must be included in the software release and [...] the license must allow modifications, derivative works and sharing [...]”

“Open science infrastructures should be [...] essentially not-for-profit and long-term”

EOSC SIRS report: Software Source Code and Open Science, 2020

connect scholarly ecosystem via Software Heritage

And much more

Software track in OSEC 2022, Software working group launched in Science Europe, DFG adds software to model CV (9/22), NASA unveils Open Science policy (12/22), ...

French National plan for Open Science, 2021-2024

SECOND FRENCH PLAN FOR OPEN SCIENCE

Generalising open science in France 2021-2024



Second French Plan for Open Science

Launch on 6 July 2021 by Frédérique Vidal, Minister for Higher Education, Research and Innovation

- Multiplying the **levers for change** in order to **generalise open science practices**
- Structuring the **policy for opening up or sharing research data**
- New commitments to the **opening of source code** produced by research
- **European and international inclusion** in the context of the French Presidency of the European Union
- **Disciplinary and thematic variations**: open science policies must be adapted to disciplinary specificities

GENERALISING
OPEN SCIENCE
IN FRANCE 2021-2024

Path Three : Opening up and promoting source code produced by research

7

Recognize and support the dissemination under an open source license of software produced by publicly funded research programmes

« The opening of software source code is a major challenge for the **reproducibility** of scientific results. »

8

Highlight the production of **source code** from higher education, research and innovation

« Distribution of software products under **open source licence** will be preferred. »

9

Define and promote an **open source software policy**

Define and promote an open source software policy

- Produce a **National Charter for Open Source Software** coming from higher education, research and innovation
- Develop the **link between data and software** through a network of **Chief Data Officers** in the various universities and research performing organisations.
- Develop the **economic models of open source software** and make them known within commercialization services
- **Support Software Heritage** and recommend it for the archiving and referencing of source code

Recognise source code as a contribution to research

- Create an **open source research software prize**
- **Provide greater recognition** for software production in the career of researchers, research support staff

Build an ecosystem that connects code, data and publications

- Develop **proper coordination** between software forges, open publication archives, data repositories and the scientific publishing sector.

ARDC

- **Archive** for retrieval (*reproducibility*)
- **Reference** for identification (*reproducibility*)
- **Describe** for discovery and reuse
- **Cite/Credit** for credit and evaluation

Before ARDC

- **Development** practices and tools (VCS, build system, test suites, CI, code quality, ...)
- **Opening up** towards a community (documentation, organization, communication)

Need training, tooling, infrastructures, best practices

Beyond ARDC

- **Policies** (dissemination, reuse, careers, ...)
- **Sustainability** (legal, financial, etc.)
- Technology transfer
- Advanced technologies and tools (quality, traceability, etc.)

let's focus on the ARDC part

How do we tackle preservation?

A - Since the 1970's 1990's

.zip or .tar file on:

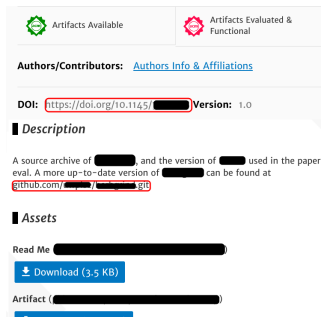
- ftp-server (e.g. [gnu](#))
- web page ([example](#))
- document archive (+ DOI [sample](#))

B - Since the 2000's

Rely on *software forges*

- institutional/project (e.g. [example](#))
- free commercial ones: BitBucket, GitHub, GitLab, ... (e.g. [parmap](#))

C: a mix of the two



Can get no satisfaction...

A *Poor user experience*

B *No preservation guarantee*

C *Can do so much better*

Forges are *not* archives!

2015: the first big bad news

Google Code and Gitorious.org shutdown: ~1M endangered repositories

- broken links in the web of knowledge (my papers too)

Big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250.000+ repositories (including research software)
- summer 2022: GitLab.com considers erasing **all** projects that are **inactive for a year**

In Academia too!

- 2021: Inria's old gforge is unplugged... **breaks the Opam build chain** for OCaml

We need a universal archive of software source code: now we have one!

- 1 Introduction
- 2 (Open) Source for Open Science
- 3 Software Heritage in the big picture
- 4 Demo time
- 5 Much more than an archive
- 6 Conclusion

Unveiled in 2016



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 and share all
software source code

Research infrastructure



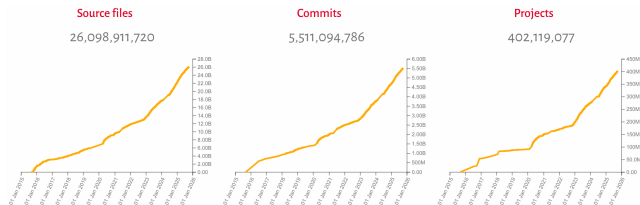
enable analysis of all
software source code

A universal software archive, as a shared infrastructure

One infrastructure
open and shared



The largest archive ever built



Bitbucket 2,818,626 origins	debian 142,984 origins	git 33,472 origins
GitHub 266,235,919 origins	Gogs 422 origins	GitLab 5,803,610 origins
git 3,824 origins	GNU 354 origins	heptapod 1,358 origins
Guix 56,248 origins	Maven 423,606 origins	NixOS 34,420 origins
launchpad 654,759 origins	npm 4,070,945 origins	Packagist 380,156 origins
PAGURE 72,459 origins	Phabricator 198 origins	pub.dev 63,003 origins
pypi 621,919 origins	SOURCEFORGE 382,368 origins	stagit 343 origins

Sharing the vision



And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors



Diamond sponsors



Platinum sponsors



Gold sponsors



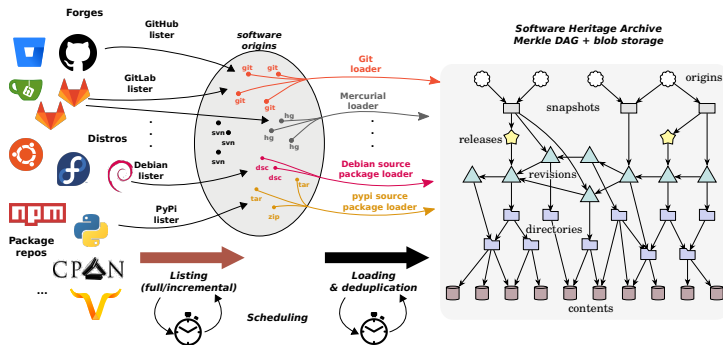
Silver sponsors



Bronze sponsors



The archive under the hood



Global development history permanently archived in a uniform data model

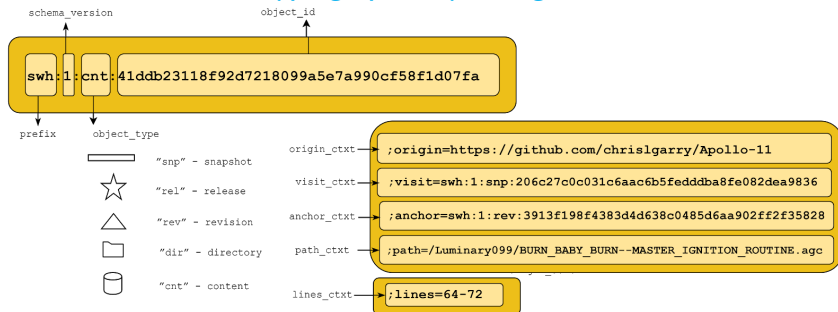
- over 24 billion unique source files from over 375 million software projects
- ~2PB (compressed) blobs, ~50 B nodes, ~900 B edges

The Software Hash persistent identifier (SWHID)

Software Hash Identifiers (SWHID)

see swhid.org

50+B intrinsic, decentralised, cryptographically strong identifiers, SWHIDs



In [SPDX 2.2](#); IANA "swh:"; WikiData [P6138](#); standard [ISO 18670](#)

Full fledged *source code references* for traceability, integrity and reproducibility

Examples: [Apollo 11 AGC](#), [Quake III rsqrt](#); Guidelines available: [HOWTO](#) and [ICMS 2020](#)

- 1 Introduction
- 2 (Open) Source for Open Science
- 3 Software Heritage in the big picture
- 4 Demo time**
- 5 Much more than an archive
- 6 Conclusion

A walkthrough

- Browse (e.g. [Apollo 11 \[excerpt\]](#), your work [may be already there](#) !)
- Trigger archival, use the [updateswh](#) browser extension, configure the webhooks
- Get and use SWHIDs ([full specification available online](#))
- Cite software with [biblatex-software](#) package from CTAN
 - [Overleaf ACMART template](#) available
- Example in journals: [article from IPOL](#)
- Example with Parmap: [devel on Github](#), [archive in SWH](#), [curated deposit in HAL](#)
- Extracting all the software products [for Inria](#), [for CNRS](#), [for CNES](#), [for LIRMM](#) or [for Rémi Gribonval](#) using [HalTools](#)
- Curated deposit in SWH via HAL, see for example: [LinBox](#), [SLALOM](#), [Givaro](#), [NS2DDV](#), [SumGra](#), [Coq proof](#), ...
- Example use in research articles:
 - compare Fig. 1 and conclusions in [the 2012 version](#) and [the updated version](#)
 - SWHID in [a replication experiment](#)

Addressing ARDC: an example is worth 1000 words

Software metadata: codemeta.json

- example from Parmap, created with Codemeta generator
- software citation (see detailed HOWTO) and biblatex-software

Integration with the HAL national french open access archive

- Curated deposit: metadata quality due to moderation
- examples: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, ...
- generation of reports, cv, web pages: for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools

Software Heritage + a *curated* metadata repository allows to address all needs ...

- *researcher, engineer*: archival, reference, credit, CV etc. *with a little effort from them*
- *labs, organizations*: track and report software production, *curated* catalog

Selected adoption indicators: users

From [Melissa Harrison's OSEC 2022 talk](#)



What are they “referencing”?

source	n	percentage
Not available	2868	46.22
GitHub	1151	18.55
software heritage	387	6.24
zenodo	142	2.29
r package	70	1.13
cran	56	0.90
r package version	54	0.87
gitlab	35	0.56

- 6205 “software” references identified
- Top 8 listed, then long tail of 1055 other sites – 932 are unique “source”

Use on replicabilitystamp.org

Lightweight Curvature Estimation on Point Clouds
with Randomized Corrected Curvature Measures

Jacques-Olivier Lachaud, David Coeurjolly, Céline Labart, Pascal Romon,
Boris Thibert
Wiley Computer Graphics Forum (CGF)



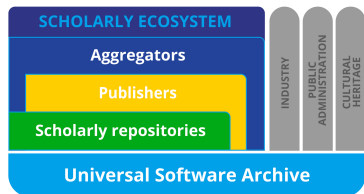
HAL+SWH in the [Open Science software booklet](#)

Funding agencies recommendations [ANR 2023 guidelines](#) (p. 17)

Enfin, conformément au 2^{ème} Plan national pour la science ouverte, L'ANR recommande que les logiciels développés durant le projet soient mis à disposition sous une [licence libre](#)³⁰ et que les codes sources soient stockés dans l'archive Software Heritage³¹ en indiquant la référence au financement ANR.

Selected adoption indicators: infrastructures

EOSC SIRS report: Software Source Code and Open Science, 2020



Connect scholarly ecosystem with the whole software ecosystem

See e.g. [the French public administration open source catalog](#)



Ongoing work: FAIRCORE4EOSC

A full workpackage:

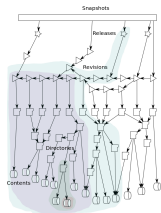
- connectors with InvenioRDM (Zenodo), episcience, Dagstuhl, sw/Math, etc.
- Software Heritage mirror for the European Open Science Cloud (EOSC)
- standardisation of CodeMeta and SWHID

- 1 Introduction
- 2 (Open) Source for Open Science
- 3 Software Heritage in the big picture
- 4 Demo time
- 5 Much more than an archive**
- 6 Conclusion

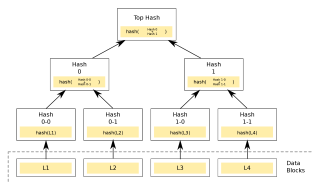
A revolutionary infrastructure

Modern "Library of Alexandria", *international, non profit, long term* initiative
addressing the needs of *industry, research, culture and society as a whole*

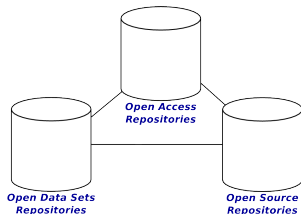
Software Graph



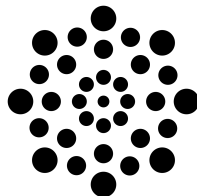
Software Blockchain



Open Science pillar



Big Code



One infrastructure, shared: more efficient, less waste ...

... for AI, Cybersecurity, Public Administration, Software Engineering ... see [UNESCO Symposium 2025](#)

Opportunity: your Institutional Portal!

Monitor and showcase your institution's software production



DALL-E's view of an institutional portal

- Curated metadata under institutional authority
- Persistent reference
- Uniform citation
- Automated extraction of reports and indicators

Why SWH?

- platform agnostic, metadata from multiple institutions
- benefit from Software Heritage's future developments

Get involved in the portal specification!

- curation workflow
 - researcher initiated (swhid deposit) vs institution initiated (metadata deposit)
- product deployment: on premise vs SaaS
- design of reports and extraction formats

- 1 Introduction
- 2 (Open) Source for Open Science
- 3 Software Heritage in the big picture
- 4 Demo time
- 5 Much more than an archive
- 6 Conclusion

Software Heritage is

- **vendor neutral**, **open source**
- **worldwide**, **long term** initiative

Software Heritage enables

- **archival** and **reference** for reproducibility
- **cost mutualisation**, **global knowledge base**

Call to action

- **show it** to students, engineers, researchers: they will adopt it
- **include** Software Heritage in your Open Science policy
- **integrate it** with your academic infrastructure
- **join** Software Heritage, support sustainability, steer development
- **get involved** with the upcoming OSPO-RADAR project

Join the Software Heritage **Archives and Libraries Interest Group**
softwareheritage.org/support/members-alig/

Appendix

- 7 Hands-on examples
- 8 Background on key concepts on identifiers
- 9 Extrinsic identifiers are fragile

Get Swhid for a code snippet in the archive

The screenshot shows the Software Heritage website in a web browser. The browser's address bar displays `https://archive.softwareheritage.org`. The website's header includes a navigation menu with links for Home, Development, Documentation, and a red 'Donate' button. A 'Full width' toggle is visible on the left. The main content area is titled 'Welcome to the Software Heritage archive'. On the left sidebar, there is a 'Features' section with links for Search, Downloads, Save code now, Add forge now, and Help. The central 'Search' section contains a text input field with the placeholder 'Enter a SWHID to resolve or string pattern(s) to search for in origin urls', a search button, and three checked filters: 'only show origins visited at least once', 'filter out origins with no archived content', and 'search in metadata (instead of URL)'. A 'visit type' dropdown menu is set to 'any'. Below the search section, the 'Overview' section explains the initiative's goal to collect, preserve, and share publicly available software. The 'Content' section mentions the 'Archive Changelog' and the 'Regular crawling' section describes how software origins are discovered and archived using 'listers'.

File Edit View History Bookmarks Tools Help

Welcome to the x +

https://archive.softwareheritage.org 150% ☆

Full width Home Development Documentation Donate Operational login

SoftwareHeritage Archive

Features

- Search
- Downloads
- Save code now
- Add forge now
- Help

Welcome to the Software Heritage archive

Search

Enter a SWHID to resolve or string pattern(s) to search for in origin urls

- ☒ only show origins visited at least once
- ☒ filter out origins with no archived content
- ☐ search in metadata (instead of URL)

visit type any

... or check our [Web API](#)

Overview

The long term goal of the Software Heritage initiative is to **collect** all publicly available software in source code form together with its development history, replicate it massively to ensure its **preservation**, and **share** it with everyone who needs it. The Software Heritage archive is growing over time as we crawl new source code from software projects and development forges.

Content

A significant amount of source code has already been ingested in the Software Heritage archive, see the [Archive Changelog](#) for notable changes to the archive over time. It currently includes the following software origins.

Regular crawling

These software origins get continuously discovered and archived using the [listers](#) implemented by Software Heritage.

video

Parmap_code_swh-poster.png

Get a citation for an archived code

 **Browse the archive**

Enter a SWHID to resolve or keyword(s) to search for 

 <https://github.com/rdicosmo/parmap> 

07 February 2025, 10:52:27 UTC

<> Code

 Branches (52)

 Releases (10)

 Visits

★ Branch: HEAD ▾ bc7ddd6 /

 History

 Download ▾

 Save again

 Extrinsic metadata

 Tip revision: **ecd3744ed558da4ea2bf9eb87b80b8949f417126** authored by Roberto Di Cosmo on 14 November 2024, 11:24:50 UTC
Merge pull request #115 from anlambert/codemeta-fox-orcid-urls

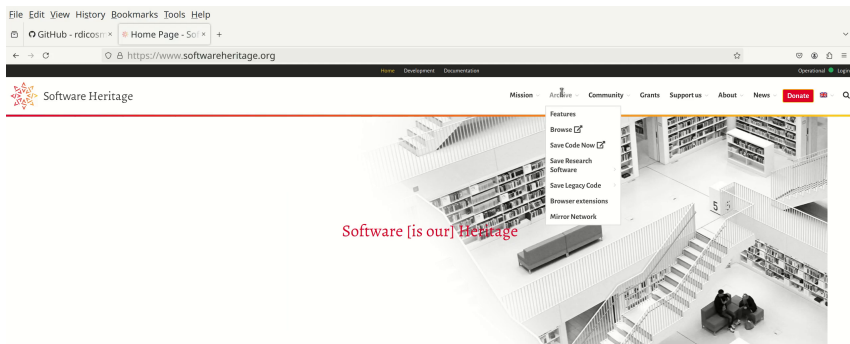
File	Mode	Size
config		
example		
src		
tests		
 .gitignore	-rW-r--r--	38 bytes
 AUTHORS	-rW-r--r--	722 bytes
 CHANGES	-rW-r--r--	1.8 KB
 LICENSE	-rW-r--r--	25.8 KB
 Makefile	-rW-r--r--	439 bytes
 README.md	-rW-r--r--	7.1 KB
 codemeta.json	-rW-r--r--	1.5 KB

 Citations

 Permalinks

video

Use the updateswh browser extension



We are building the universal software archive



Collect

We **collect** and **preserve** software in source code form, because software embodies our technical and scientific knowledge and humanity cannot afford the risk of losing it.

Software is a precious part of our cultural heritage. We curate and make accessible all the software we collect, because only by **sharing** it we can guarantee its preservation in the very long term.

[Browse the archive](#)

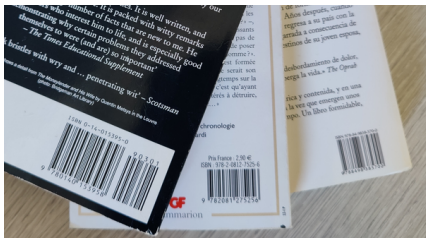
[Discover our mission](#)



video

- 7 Hands-on examples
- 8 Background on key concepts on identifiers**
- 9 Extrinsic identifiers are fragile

Identification of a book



Goal: identify a book

- one ISBN number per published book
- ISO 2108 standard specification

Location of (a copy of) a book



Goal: find (a copy of) a book

- many locations (locations can change!)
- many approaches for call numbers

we are interested in **identification**, not in location

Extrinsic vs Intrinsic identifiers

In a nutshell

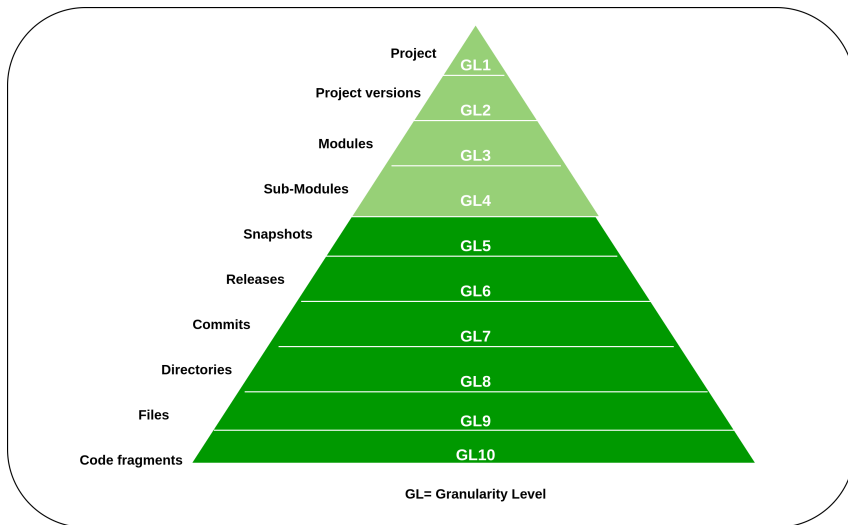
(for more info see [this dedicated blog post](#))

Main difference: how the *relation* between *identifier* and *designated object* is created and maintained. *Persistence* is a key desired property.

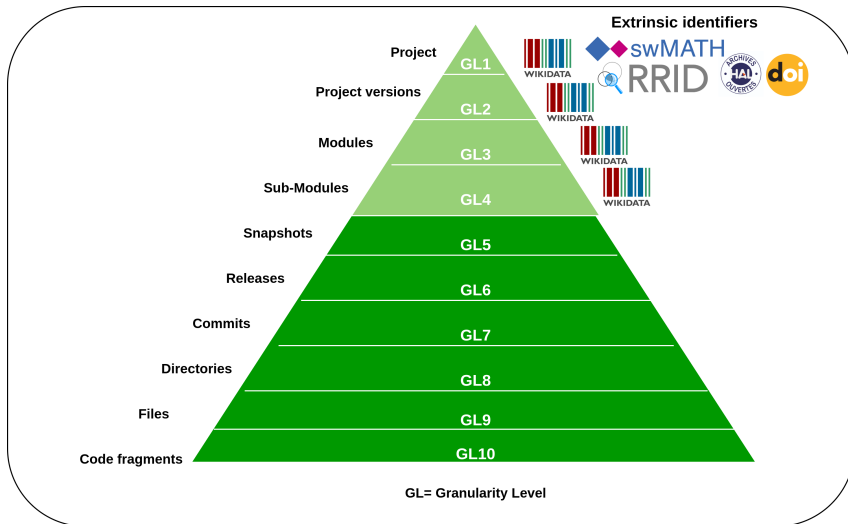
	Extrinsic	Intrinsic
relation	register	convention
persistence	external ¹	internal
pre-internet	passport number, ISBN, SSN, etc.	Music/Chemistry notations <i>e.g. NaCl is table salt</i>
internet era	DOI, Handle, Ark, etc.	cryptographic hashes <i>e.g.: git, bitcoin, SWHID</i>

Software development has *massively adopted* intrinsic identifiers *since 2006*
Git, GitHub, GitLab, pull requests, Guix, Nix, etc. *all rely on* intrinsic identifiers
replicability/reproducibility need intrinsic identifiers
SWHID is *directly compatible* with git!

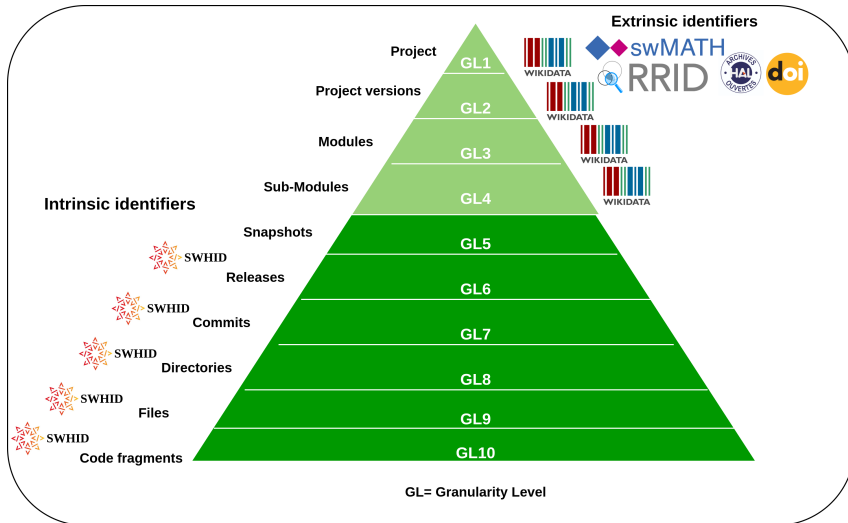
¹"persistence... is a function of *administrative care*" [RFC 3650 \(Handle System Overview, 2003\)](#)



Top concept layers vs. bottom artifact layers

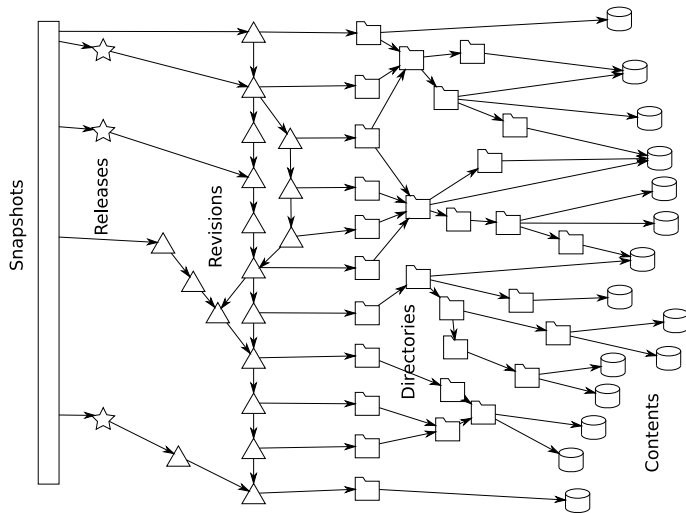


Extrinsic identifiers are key for the concept layers

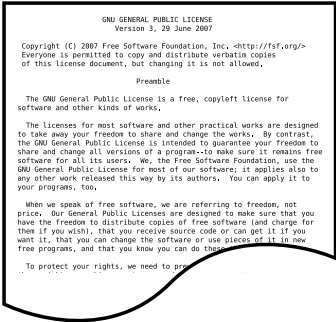


Intrinsic identifiers are key for the artifact layers

SWHID computation: 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,

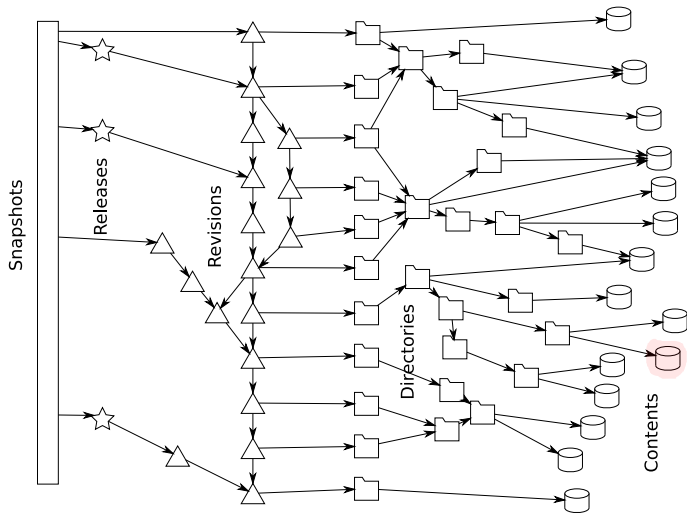
The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all 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 others from denying you
these rights or asking you to pay for these rights. This we can do only by
making the program a copyleft program, which means that whoever
distributes copies of the program must also give you the rights that are
assured by this license.

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

SWHID computation: a worked example





Directories

?

SWHID computation: a worked example

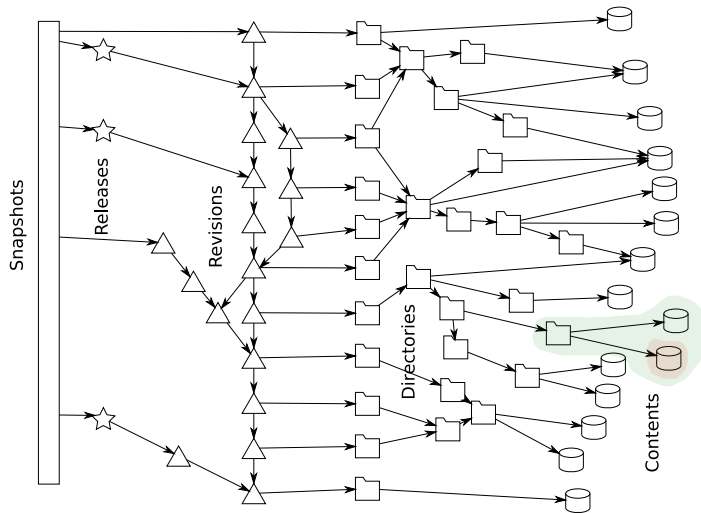


Directories

```
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

SWHID computation: a worked example



Revisions

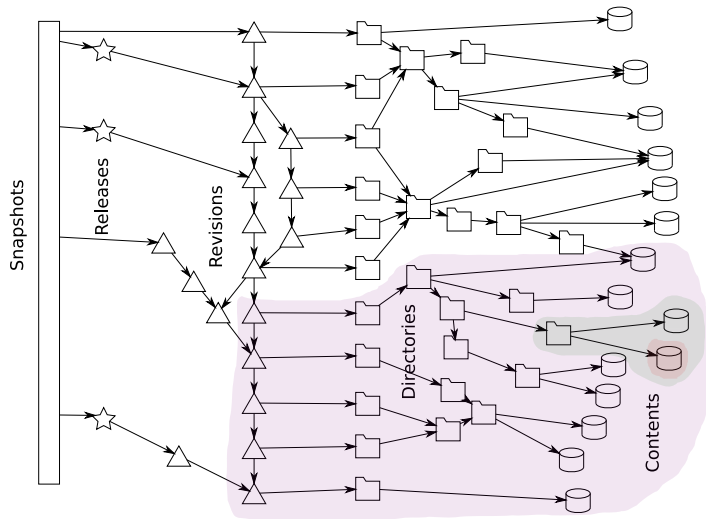
Details	Changes	Files
SHA: 963634dca6ba5dc37e3ee426ba091092c267f9f6		
Author: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Committer: Nicolas Dandrimont <nicolas@dandrimont.eu> (Thu Sep 1 14:26:13 2016)		
Subject: provenance.tasks: add the revision -> origin cache task		
Parent: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

SWHID computation: 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

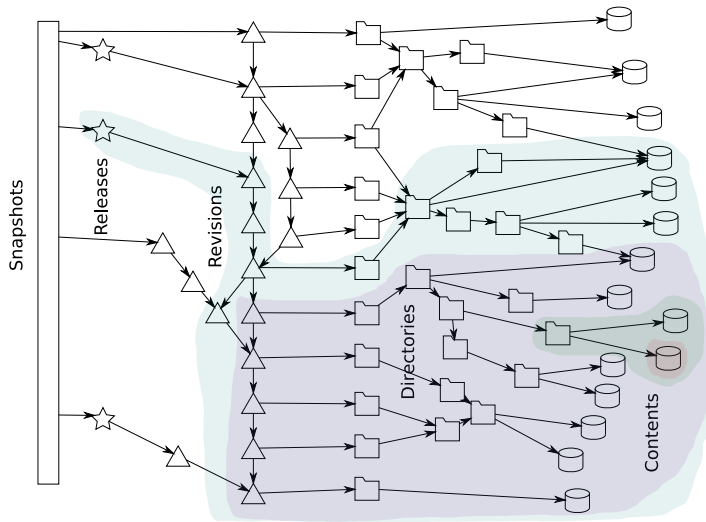
- Add new metadata column to origin_visit
- Update swh-add-directory script for updated API

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

iQIzBAABCAAdBQJXvZTNFhXuaWNvbGFzQGRhbmRyaW1vbnQuZXUACgkQ7AWLMo2+
neqorw/aaq6SOb5DijzEa+kWN3rXgVS+1K1vEVh1wNKAwx8eKJ7aX2kEiLdt7uf
ahpZ6pz3q8nqs6aC1+YrxBfcih3L2YtrdZeWXWqr8xWNMaEoYDb8qaphwh8AD5t2
ICBlit2ujtXuCrDt93eKKPvzZXg+h80sMWy3SDr6jW7Z7K4Mu/PgGlyIHPY55yo
IGEndWno7Vfh1Vm6t1n5qb7l5mXRagA+becqddubTZ2xij+jpIUqC8cyqN3hm/fL
qsJ2mu8kyz3t8tG/H1/pV+I5OwBlNpO5S5TH0tuojoEVgPK/dHSP79QuHDHfZfCao
kij6kAWyU80Mxb+nKVjjeLbrR3+yWBFj3Qp5a1/V8oOTd6E1dALcNMpEaKCoKtMt
d/gMRax1I1/g0EDfnsW67G6sDwKPKPhgFVLQ3nV3GaQQTnu1RpMz006H9/tAwzC
Gg/K1PdHT4hz0iI46wYPZyje0U2VXGFu6vU9vFQ4ZR/Wjn+0zMzdcRdrIJSUOMn
RpTTFUsbXUeXHGOpkgXhSYTnvp1gdPc76USTsK0aGe84AZm1Ik0mGrwXCVfPqJyO
nhhibB5HBNMoqyF6yTSOpUbyK70tpYRRUGKWDerK0wKSxkWLUZGtKzy6jYqJJo29
gulwGZQif5qWQCBOontAL2+HvPfaVyckMejUhg62cP/+EHlvUk=
=kOxp
-----END PGP SIGNATURE-----

id: 85083a5cc14a441c89dea73f5bdf67c3f9c6afdb

SWHID computation: a worked example



git show-refs

Snapshots

```
commit 08ffeb25770109525eb3ce21691466c53a1d9158 refs/heads/atime
commit ba5443a24e3f9fe323a46c292cec4fcb61c67eb refs/heads/directory-listing-arrays
commit d69e0dbf892383ff6589b27fbelc05d27238d9c5 refs/heads/foo
commit cf7ff9eeaa0eb22f8946908f5a8019f67de468e08 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 3590e0ca0ebbb070e5b376705fa230bbfa4ffa5cc 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 606979a4ca05d497fc0d24aad0dce82636ef47c refs/tags/v0.0.19
tag 32bf5a59fc2a323baa6d5f15a6ad5382ec275a67 refs/tags/v0.0.2
tag 3147c3d31ec46cf6492f881e908b1237ebdff2c7 refs/tags/v0.0.20
tag 215ea50daba11e082e0b72e76eb4b6073a87908 refs/tags/v0.0.21
tag 3fb168c2072a5d6252124257a1e5dfc0f5ffa1df refs/tags/v0.0.22
tag 8cdbee8da4d73fc5d262789e460a16ac3c72aba4 refs/tags/v0.0.23
...
```

id: b464cad1b66fff266a37b46ea6e7a04b545e904b

- 7 Hands-on examples
- 8 Background on key concepts on identifiers
- 9 Extrinsic identifiers are fragile**

Disruption of 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

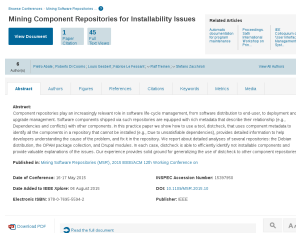
- the *half-life* of a referenced URL *is approximately 4 years* from its publication date.
D. Spinellis. The Decay and Failures of URL References.
Communications of the ACM, 46(1):71-77, January 2003.

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

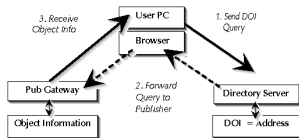
The Digital Object Identifier (DOI)

Example: <https://doi.org/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 ...



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*