

Software Heritage

Source Code Archival and Analysis at the Scale of the World

Stefano Zacchioli

Software Heritage – zack@upsilon.cc, [@zacchiro](https://twitter.com/zacchiro)

10 October 2019

Journées Calcul et Données — Toulouse, France



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Software is everywhere in modern research



[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data

Christine Borgman, Paris, 2018



Software Source code: pillar of Open Science

Software is everywhere in modern research



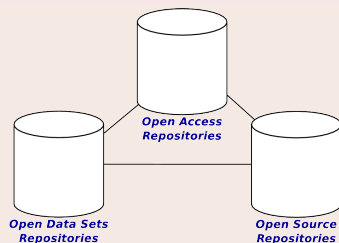
[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data

Christine Borgman, Paris, 2018

Open Science: three pillars



Software Source code: pillar of Open Science

Software is everywhere in modern research



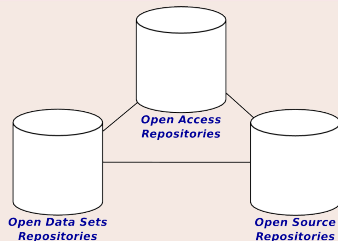
[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data

Christine Borgman, Paris, 2018

Open Science: three pillars



Nota bene

The links in the picture are **essential**

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)

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

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

    return y;
}
```

Net. queue in Linux (excerpt)

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

/* SFB 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.”

The state of the art is not ideal

Analysis of 613 papers

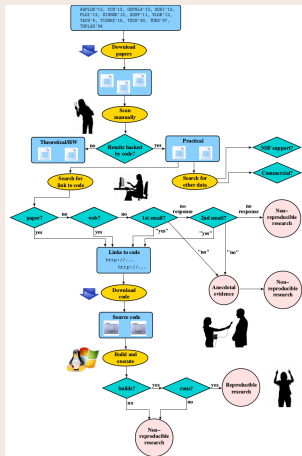
- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

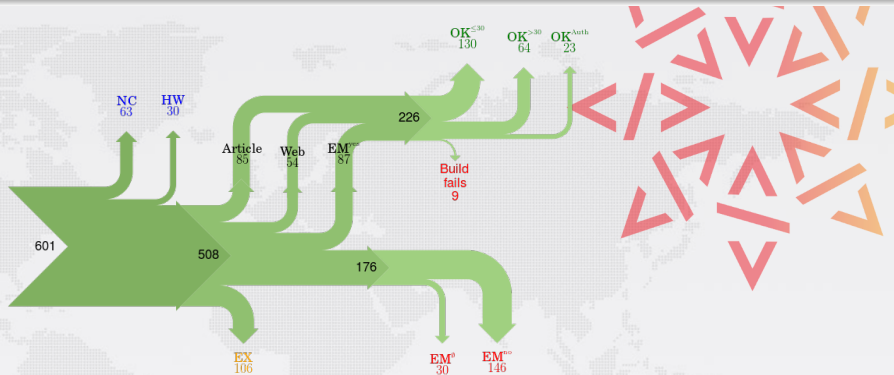
all very practical oriented

The basic question

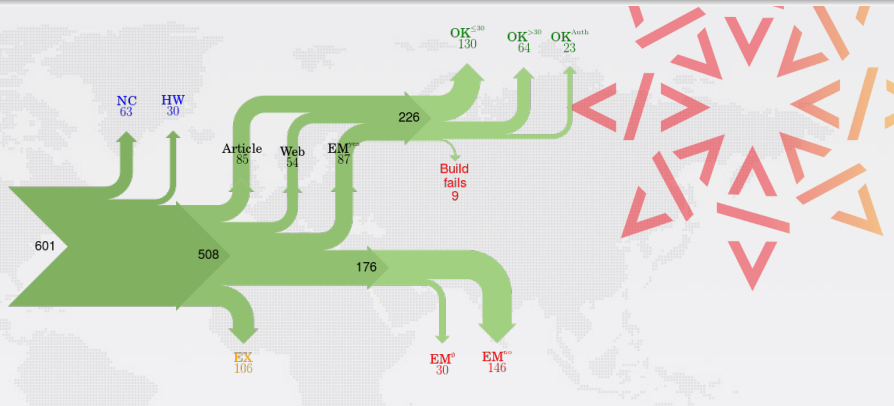
can we get the code to build and run?

The workflow





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



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

The main reasons

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

Software is fragile



A word cloud centered on the slide, featuring terms related to software fragility. The words are arranged in a circular pattern, with 'damage' and 'disaster' being the largest. Other visible words include 'malicious', 'attack', 'obsolete', 'dependencies', 'deletion', 'reference', 'storage', 'dangling', 'wear', 'corruption', 'encryption', 'format', 'aging', 'media', and 'tear'.

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code)
- for obsolete code: physical media decay (data rot)

Software is fragile



A word cloud centered on the slide, featuring terms related to software fragility. The words are arranged in a circular pattern, with 'damage' and 'disaster' being the largest. Other visible words include 'malicious', 'obsolete', 'attack', 'dependencies', 'deletion', 'reference', 'storage', 'dangling', 'wear', 'corruption', 'encryption', 'format', 'media', 'aging', and 'tear'. The background of the slide shows a faint world map and a decorative pattern of colorful arrows pointing outwards from the right side.

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code)
- for obsolete code: physical media decay (data rot)

Where is the archive...

where do we go if (a repository on) GitHub or GitLab.com goes away?

Software lacks its own research infrastructure



A wealth of software research on crucial issues...

- safety, security, test, verification, proof
- software engineering, software evolution
- big data, machine learning, empirical studies

Software lacks its own research infrastructure



A wealth of software research on crucial issues...

- safety, security, test, verification, proof
- software engineering, software evolution
- big data, machine learning, empirical studies

If you study the stars, you go to Atacama...

... where is the *very large telescope* of source code?



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE



Our mission

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

Past, present and future

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

Archiving goals

Targets: VCS repositories & source code releases (e.g., tarballs)

We DO archive

- file **content** (= blobs)
- **revisions** (= commits), with full metadata
- **releases** (= tags), ditto
- where (**origin**) & when (**visit**) we found any of the above

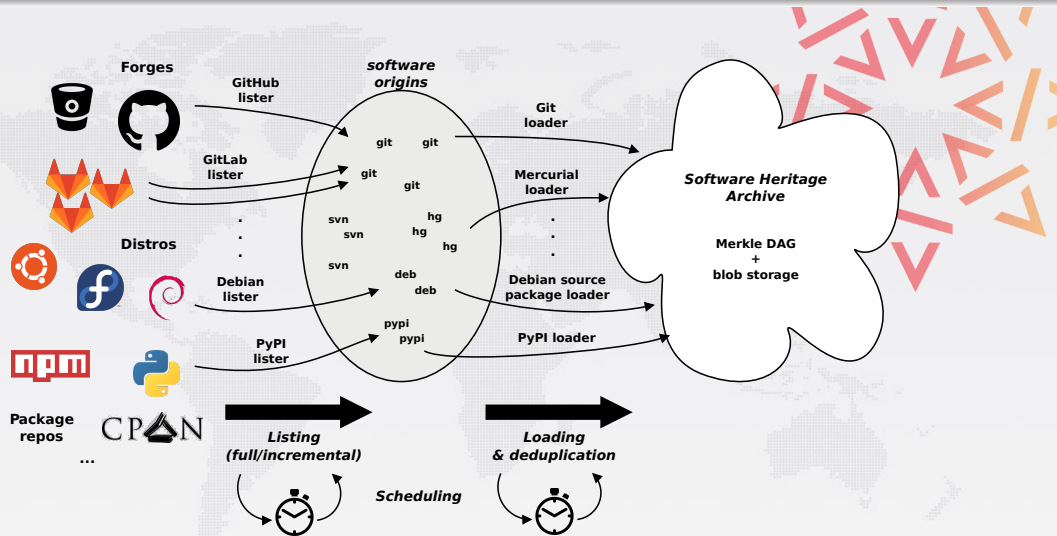
... in a VCS-/archive-agnostic **canonical data model**

We DON'T archive

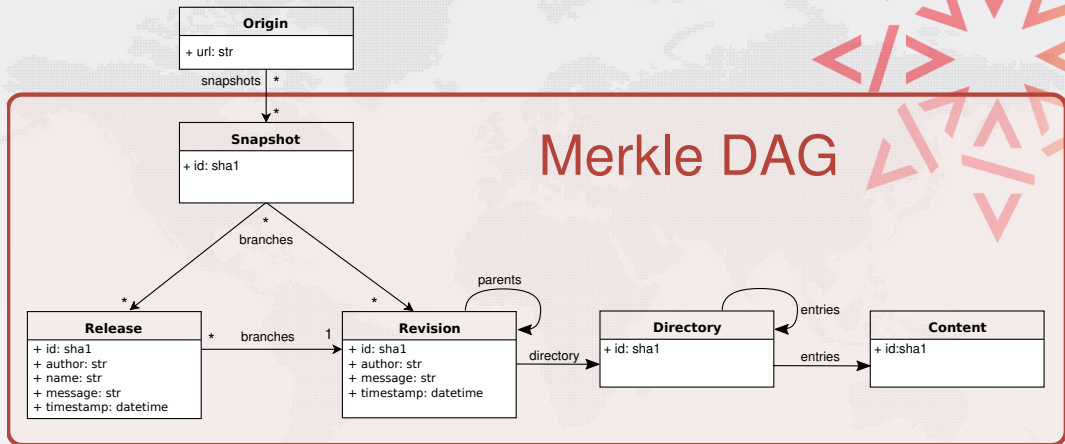
- homepages, wikis
- BTS/issues/code reviews/etc.
- mailing lists

Long term vision: play our part in a *"semantic wikipedia of software"*

Data flow



The archive: a (giant) Merkle DAG



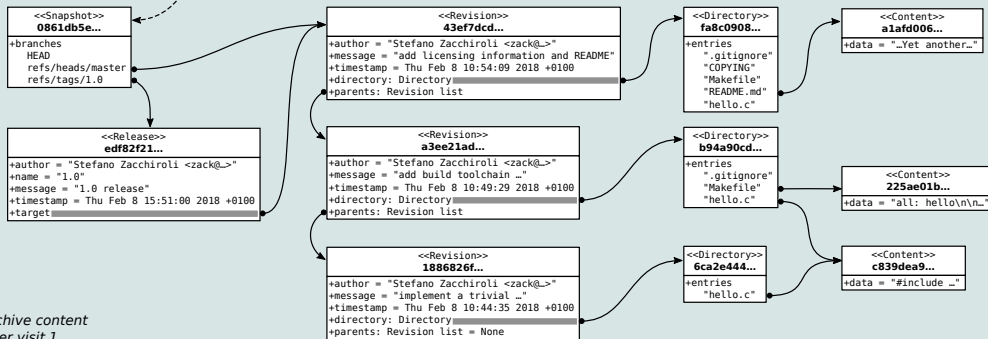
The archive: a (giant) Merkle DAG

origin
https://forge.softwareheritage.org/source/helloworld.git

visit
1

snapshot
0861db5e...

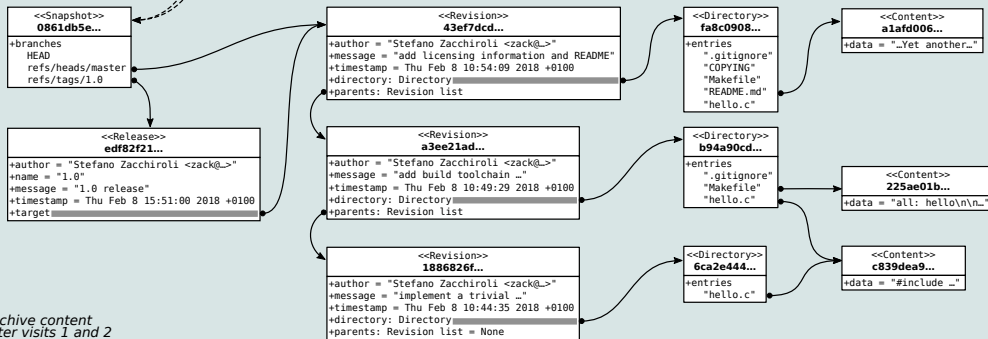
timestamp
Fri Feb 9 12:38:45 2018 +0100



Archive content
after visit 1

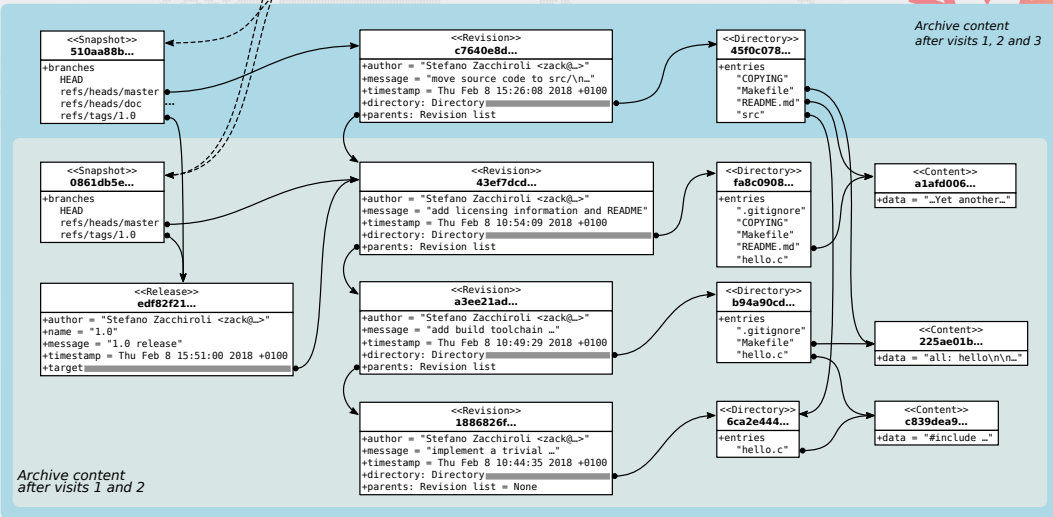
The archive: a (giant) Merkle DAG

origin	visit	snapshot	timestamp
https://forge.softwareheritage.org/source/helloworld.git	1	0861db5e...	Fri Feb 9 12:38:45 2018 +0100
https://forge.softwareheritage.org/source/helloworld.git	2	0861db5e...	Fri Feb 9 13:29:00 2018 +0100



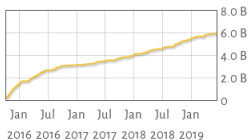
The archive: a (giant) Merkle DAG

origin	visit	snapshot	timestamp
https://forge.softwareheritage.org/source/helloworld.git	1	0861db5e...	Fri Feb 9 12:38:45 2018 +0100
https://forge.softwareheritage.org/source/helloworld.git	2	0861db5e...	Fri Feb 9 13:29:00 2018 +0100
https://forge.softwareheritage.org/source/helloworld.git	3	510aa88b...	Fri Feb 9 15:52:50 2018 +0100



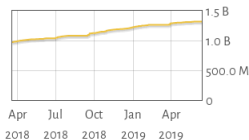
Source files

6,006,503,960



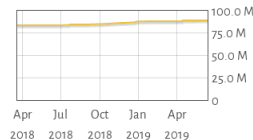
Commits

1,326,776,432



Projects

89,301,694



GitHub

debian



GitLab



Google code



GITORIOUS



GNU

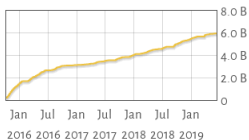
HAL
archives-ouvertes.fr

Inria
inventeurs du monde numérique

python
Package Index

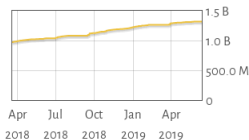
Source files

6,006,503,960



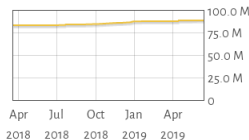
Commits

1,326,776,432



Projects

89,301,694



GitHub

debian



GitLab



Google code



GITORIOUS



GNU

HAL
archives-ouvertes.fr

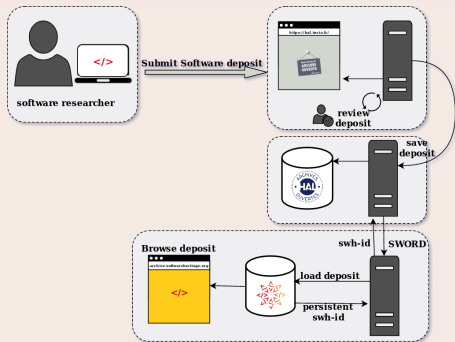
Inria
inventeurs du monde numérique

python
Package Index

- ~400 TB (uncompressed) blobs, ~20 B nodes, ~280 B edges
- The *richest* public source code archive, ... and growing daily!

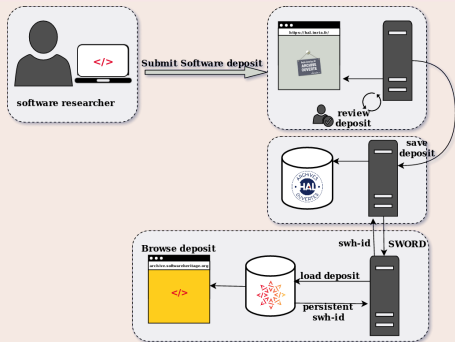
Deposit research software

open since 9/2018



Deposit research software

open since 9/2018

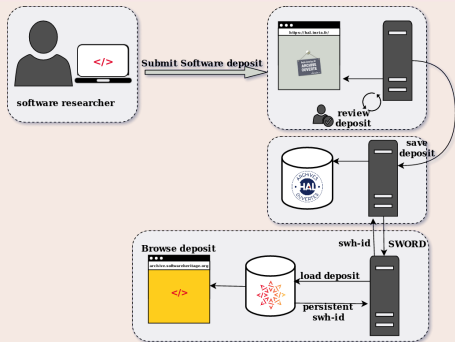


Generic mechanism (SWORD based):

- review process, versioning

Deposit research software

open since 9/2018



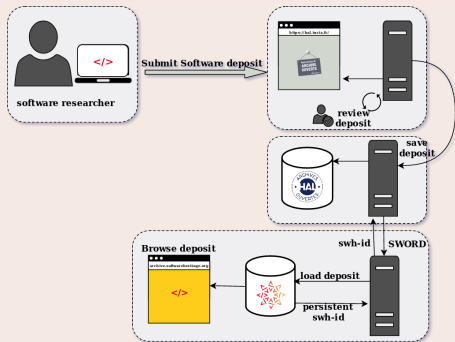
Generic mechanism (SWORD based):

- review process, versioning
- (today): deposit .zip or .tar.gz file (*guide*)
- (tomorrow): provide *SWH id* and (extract) metadata

[click here to learn more...](#)

Deposit research software

open since 9/2018



Generic mechanism (SWORD based):

- **review process**, versioning
- **(today)**: deposit .zip or .tar.gz file (*guide*)
- **(tomorrow)**: provide *SWH id* and (extract) metadata

[click here to learn more...](#)

Reference archive: origins

swMATH.org links into Software Heritage

- e.g. *the SemiPar entry in swMATH.org*

Reference archive: releases

Wikidata *SWH Release ID Property*

- e.g. *the release 3.1.0 of Gensim*

Leveraging Software Heritage for Open Science (cont.)

GNU Guix

April 18, 2019

Software Heritage and GNU Guix join forces
to enable long term reproducibility



code.etalab.gouv.fr



code.etalab.gouv.fr (alpha)

Contact Glossaire À propos

Codes sources du secteur public

Ce site vous permet de parcourir une partie des codes sources ouverts par des organismes publics. Si vos codes sources ne sont pas référencés, [envoyez-nous](#) les liens vers vos dépôts.

Dépôts de code source

Organisations ou groupes

Chiffres

Recherche libre

Licence

Langage

☐ Fourches seules

☐ Sauf archives

☐ Avec description

☐ Avec licence identifiée

2464 dépôts

1/1

1/1

Organisation / dépôt	Archive	Description	Maj	Fourches	Étoiles	Tickets
snosan-tools / avurnavs-fichiers	✱	Historique des avis urgents aux navigateurs en France métropolitaine en CSV	10/9/2019	0	1	0
OpenCTI-Platform / opentcti	✱	Open Cyber Threat Intelligence Platform	10/9/2019	98	776	84
fabnumdef / e-chauffeur_user	✱		10/9/2019	0	0	0
fabnumdef / e-chauffeur_dashboard	✱		10/9/2019	1	0	0
fabnumdef / e-chauffeur_anl	✱		10/9/2019	0	0	1

Realizing the "large telescope of source code" in practice

Requirements

- **Availability**: Software Heritage mirror, relatively up-to-date
- **Efficiency**: massive computing resources with fast access to the mirror
- **Sustainability**: pay-per-use or bring-your-own-computing

Challenges

- mirroring
- compression
- efficient processing
- experiments description language
- big code analysis (i.e., ML on source code)

... at the scale of the world!

Software Heritage Graph dataset

Use case: large scale analyses of the most comprehensive corpus on the development history of free/open source software.

Dataset

- Relational representation of the full graph as a set of tables
- Available as open data: <https://doi.org/10.5281/zenodo.2583978>

Formats

- Local use: PostgreSQL dumps, or Apache Parquet files (~1 TiB each)
- Live usage: Amazon Athena (SQL-queriable)

MSR 2020 Mining Challenge

[About](#)[Call for Papers](#)[Resources for Participants](#)

Important Dates

 AoE (UTC-12h)

Thu 30 Jan 2020
Abstracts due

Thu 6 Feb 2020
Papers due

Call for Papers

The International Conference on Mining Software Repositories (MSR) has hosted a mining challenge since 2006. With this challenge, we call upon everyone interested to apply their tools to a common dataset. The challenge is for

Wrapping up

- Software Heritage archives all software source code with its development history.
- It is a major endeavor that benefits society, science, and industry.
- For computer scientists, it is a gold mine of research opportunities. Wanna join?

References



Antoine Pietri, Diomidis Spinellis, Stefano Zacchirolì

The Software Heritage graph dataset: public software development under one roof
MSR 2019: Mining Software Repositories, IEEE



Jean-François Abramatic, Roberto Di Cosmo, Stefano Zacchirolì

Building the Universal Archive of Source Code
Communication of the ACM, October 2018



Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchirolì

Identifiers for Digital Objects: the Case of Software Source Code Preservation
iPRES 2018: Intl. Conf. on Digital Preservation



Roberto Di Cosmo, Stefano Zacchirolì

Software Heritage: Why and How to Preserve Software Source Code
iPRES 2017: Intl. Conf. on Digital Preservation