Software Heritage

The universal source code archive

Roberto Di Cosmo, Morane Guenpeter

Director, Software Heritage
Computer Science full professor, Inria and IRIF
roberto@dicosmo.org

July 11th, 2018

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline

Software Heritage in a nutshell

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

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

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

Software Heritage in a nutshell

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

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

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

Reference catalog

find and reference all the source code

Software Heritage in a nutshell

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

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

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

Reference catalog

find and reference all the source code

Universal archive

preserve all the source code

Software Heritage in a nutshell

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

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

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

Reference catalog

find and reference all the source code

Universal archive

preserve all the source code

Research infrastructure

enable analysis of all the source code

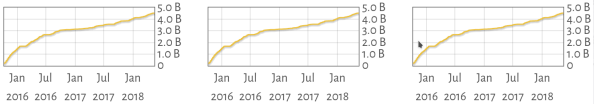
Our principles

Our principles

Cultural Heritage | **Industry** | **Research** | **Education**

Software Heritage

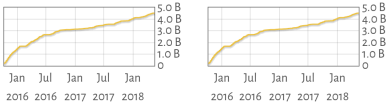
Source files	Commits	Projects
4,536,067,027	1,024,675,748	83,801,775

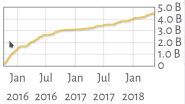


Cultural Heritage | Industry | Research | Education

Software Heritage

Source files	Commits	Projects
4,536,067,027	1,024,675,748	83,801,775





Open approach

- open source
- transparency

In for the long haul

- non profit, mirrors
- intrinsic identifiers

Exhaustive

- all software
- open to all communities

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 · INAP · THE LINUX FOUNDATION · ADULLACT · INFORMATICS EUROPE · Computer History Museum · software freedom conservancy · Software Freedom Law Center · AdaCore · gandi.net · TDF · FREE SOFTWARE FOUNDATION · creative commons · SIF · open source initiative · openinventionnetwork

Sponsoring our work

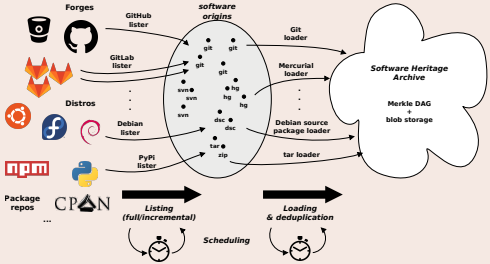
Inria — INVENTEURS DU MONDE NUMÉRIQUE

Microsoft · intel · SOCIETE GENERALE · HUAWEI · Google · DANS · NOKIA Bell Labs · GitHub · UQÀM · FOSSID

Outline

Zoom on the collection phase

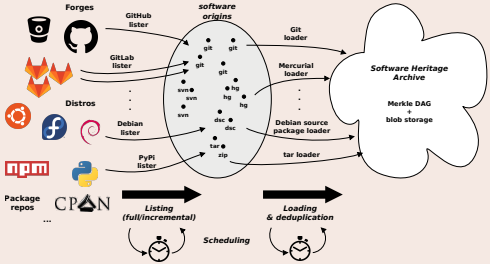
Much more than an archive!



- GitHub
- Debian, GNU
- Gitorious, Google Code
- WIP: Bitbucket, FusionForge, GitLab.com
- *add your own plugins!*

Zoom on the collection phase

Much more than an archive!



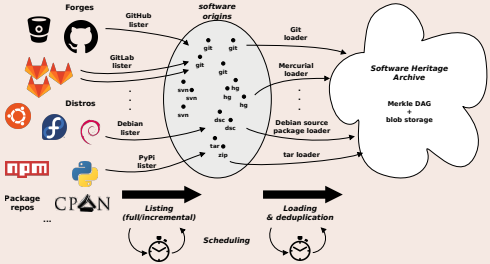
- GitHub
- Debian, GNU
- Gitorious, Google Code
- WIP: Bitbucket, FusionForge, GitLab.com
- *add your own plugins!*

Important properties

- mission: exhaustive and up to date collection of source code, *specifically*
- strategy: automatic harvesting + *deposit* from *selected* sources

Zoom on the collection phase

Much more than an archive!



- GitHub
- Debian, GNU
- Gitorious, Google Code
- WIP: Bitbucket, FusionForge, GitLab.com
- *add your own plugins!*

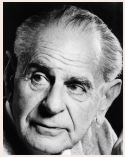
Important properties

- mission: exhaustive and up to date collection of source code, *specifically*
- strategy: automatic harvesting + *deposit* from *selected* sources

The *richest* source code archive already, … and growing daily!

How we built our scientific knowledge

Reproducibility is the key

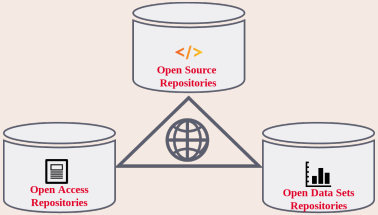


non-reproducible single occurrences are of no significance to science

Karl Popper, The Logic of Scientific Discovery, 1934

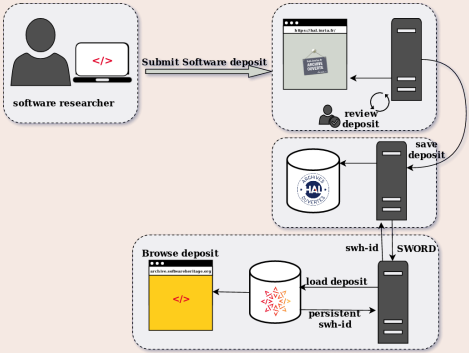
the *software source code* is special

- It embodies the logic of the data transformation.
- It must be included in strategies for *scientific knowledge preservation.*
- knowing the exact version of the software used is essential for reproducibility.



The research software (deposit) use case

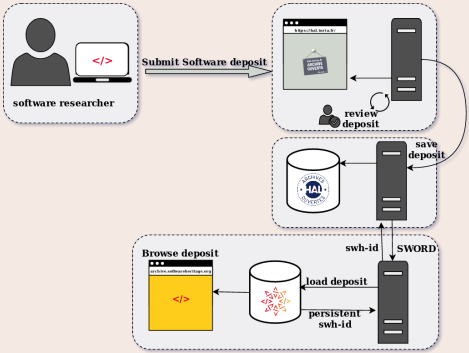
Deposit software in HAL http://hal.inria.fr/hal-01738741



The research software (deposit) use case

Deposit software in HAL

http://hal.inria.fr/hal-01738741

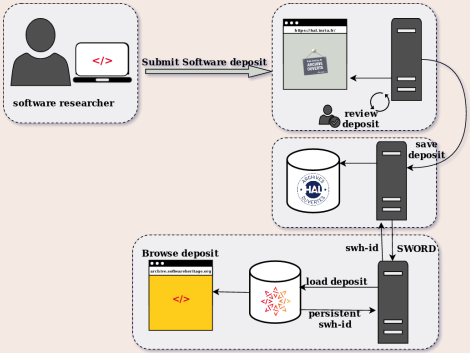


Generic mechanism:

- SWORD based
- review process
- versioning

The research software (deposit) use case

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



Generic mechanism:

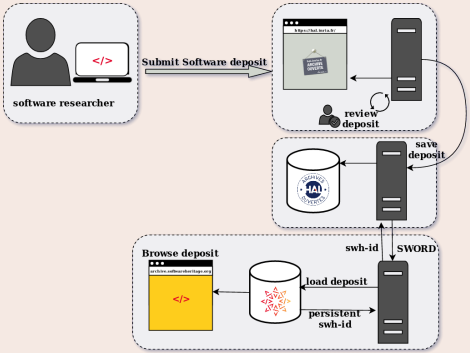
- SWORD based
- review process
- versioning

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
  - ...

The research software (deposit) use case

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/

Identifying and retrieving source code

Intrinsic identifiers (spec: `http://bit.ly/swhpids`)

- provide integrity guarantees
- all software and VCS (not just git or GitHub)
- use for identifying a precise version of source code
- learn more in the forthcoming iPres 2018 paper

e.g: swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa

Identifying and retrieving source code

Intrinsic identifiers (spec: `http://bit.ly/swhpids`)

- provide integrity guarantees
- all software and VCS (not just git or GitHub)
- use for identifying a precise version of source code
- learn more in the forthcoming iPres 2018 paper

e.g: swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa

"Wayback-machine-style" identifiers

- point to software origins
- expose the SWH crawling history

use when no precise version is known

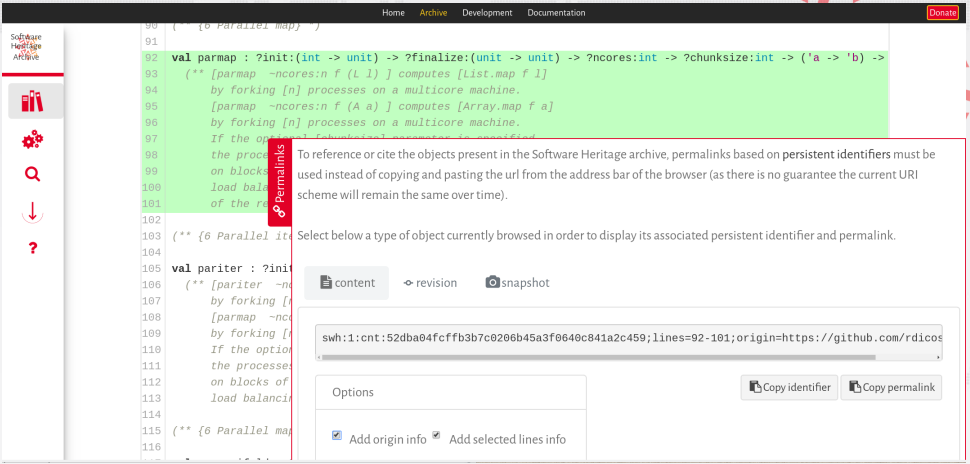
Demo Time!

- example deposits in HAL
- example use of `https://archive.softwareheritage.org`

Access using intrinsic IDs

Click on the links in the paper and view the source code!

"Our Parmap.parmap and Parmap.parfold functions may be used to seamlessly ..."



Selected unique benefits there are more!

All features of Software Heritage *for free*

- intrinsic IDs (integrity, not dependent on resolvers!)
  - specification: `http://bit.ly/swhpids`
  - iPres2018 paper: `http://bit.ly/swhpidpaper`
- browse, download (now)
- metadata, licenses, provenance (plagiarism detection), classification (wip), ...

Selected unique benefits

there are more!

All features of Software Heritage *for free*

- intrinsic IDs (integrity, not dependent on resolvers!)
  - specification: `http://bit.ly/swhpids`
  - iPres2018 paper: `http://bit.ly/swhpidpaper`
- browse, download (now)
- metadata, licenses, provenance (plagiarism detection), classification (wip), ...

Coverage and uniformity

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

Selected unique benefits

there are more!

All features of Software Heritage *for free*

- intrinsic IDs (integrity, not dependent on resolvers!)
  - specification: `http://bit.ly/swhpids`
  - iPres2018 paper: `http://bit.ly/swhpidpaper`
- browse, download (now)
- metadata, licenses, provenance (plagiarism detection), classification (wip), ...

Coverage and uniformity

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

Sustainability

... doors are open!

one infrastructure *independent* non profit foundation *worldwide* mirrors

Breaking news

Operational adoption

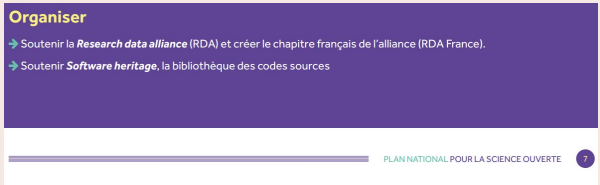
June 7th swMath.org points into SWH for the source code see "Code" link in, e.g. `http://swmath.org/software/7116`

ongoing OpenAire detects source code links in articles, resolves them to SWH

September HAL opens the software deposit doors on all portals

Institutional adoption

July 4th Software Heritage is part of the french National Plan for Open Science



Come in, we're open! (research software focus)

Adopt and reuse

deposit/reference research software in SWH	morane.gg@gmail.com
become a SWH mirror	roberto@dicosmo.org

Standardise

metadata	RDA source code IG
identifiers	RDA source code identification WG

Support

sponsoring / partnership	sponsorship.softwareheritage.org
donations	softwareheritage.org/donate
our own code	forge.softwareheritage.org

Outline

Our challenge in the PID arena

Long term

Identifiers must be there for the long term

No middle man

Identifiers must be meaningful even if resolvers go away

Integrity, not just naming

Identifier must ensure that the retrieved object is the intended one

Uniqueness by design

one name identifies a single object, and each object has only one name

Exploring the PID landscape

A lot of options out there...

URL, URI, PURL, URN, ARK, DOI, ...

... some are widely used

- articles
- data
- even software artefacts!

Exploring the PID landscape

A lot of options out there...

URL, URI, PURL, URN, ARK, DOI, ...

... some are widely used

- articles
- data
- even software artefacts!

We can get no satisfaction

of all the key criteria

Exploring the PID landscape

A lot of options out there...

URL, URI, PURL, URN, ARK, DOI, ...

... some are widely used

- articles
- data
- even software artefacts!

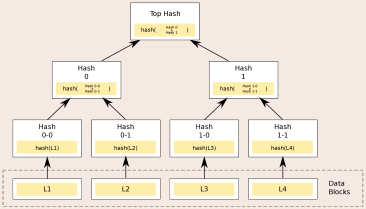
We can get no satisfaction

of all the key criteria

we adopted something radically different

Intrinsic identifiers in Software Heritage

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



Combination of

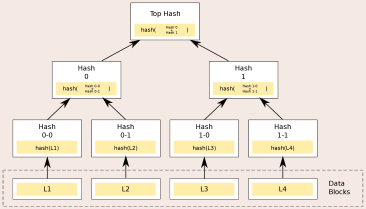
- tree
- hash function

Classical cryptographic construction

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

Intrinsic identifiers in Software Heritage

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
- widely used in industry (e.g., Git, nix, blockchains, IPFS, ...)

Back to basics: DIOs vs. IDOs

DIO (digital identifier of an object)

- digital identifiers for traditional (non digital) objects
  - epistemic complications (manifestations, versions, locations, etc.)
  - significant governance issues, ...

Back to basics: DIOs vs. IDOs

DIO (digital identifier of an object)

- digital identifiers for traditional (non digital) objects
  - epistemic complications (manifestations, versions, locations, etc.)
  - significant governance issues, ...

IDO (identifier of a digital object)

- (digital) identifier for digital objects
  - much simpler to build/handle
  - can (and must) be intrinsic

Back to basics: DIOs vs. IDOs

DIO (digital identifier of an object)

- digital identifiers for traditional (non digital) objects
  - epistemic complications (manifestations, versions, locations, etc.)
  - significant governance issues, ...

IDO (identifier of a digital object)

- (digital) identifier for digital objects
  - much simpler to build/handle
  - can (and must) be intrinsic

Separation of concerns

- yes, we need both DIOs and IDOs
- no, we must not mistake DIOs for IDOs (and viceversa)

Back to basics: DIOs vs. IDOs

DIO (digital identifier of an object)

- digital identifiers for traditional (non digital) objects
  - epistemic complications (manifestations, versions, locations, etc.)
  - significant governance issues, ...

IDO (identifier of a digital object)

- (digital) identifier for digital objects
  - much simpler to build/handle
  - can (and must) be intrinsic

Separation of concerns

- yes, we need both DIOs and IDOs
- no, we must not mistake DIOs for IDOs (and viceversa)

Working together

Example: links to *software source code* in an article

Leveraging the Software Heritage universal archive:

set of files `swh:1:dir:06741c8c37c5a384083082b99f4c5ad94cd0cd1f`
id of directory object listing all the files in a project (at a given time)

revision `swh:1:rev:7598fb94d59178d65bd8d2892c19356290f5d4e3`
id of commit object which a tree and (a pointer to) the history

Full specification available online:
`https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html`

Working together

Example: links to *software source code* in an article

Leveraging the Software Heritage universal archive:

set of files `swh:1:dir:06741c8c37c5a384083082b99f4c5ad94cd0cd1f`
id of directory object listing all the files in a project (at a given time)

revision `swh:1:rev:7598fb94d59178d65bd8d2892c19356290f5d4e3`
id of commit object which a tree and (a pointer to) the history

Full specification available online:
`https://docs.softwareheritage.org/devel/swh-model/persistent-identifiers.html`

metadata this *will* involve some form of DIO

- and we get all the complications back

Outline

Our role : handle *all* the *software source code*

At the end of the process

Explicit deposit, coordinated with the publisher

- store the *final* source code (no garbage)
- store only public source code
- N.B.: no embargo or access control (yet)

During the review

Access to the largest available source code base

- provenance, plagiarism detection (for new code)
- metrics (for long standing projects)

Our role : handle *all* the *software source code*

At the end of the process

Explicit deposit, coordinated with the publisher

- store the *final* source code (no garbage)
- store only public source code
- N.B.: no embargo or access control (yet)

During the review

Access to the largest available source code base

- provenance, plagiarism detection (for new code)
- metrics (for long standing projects)

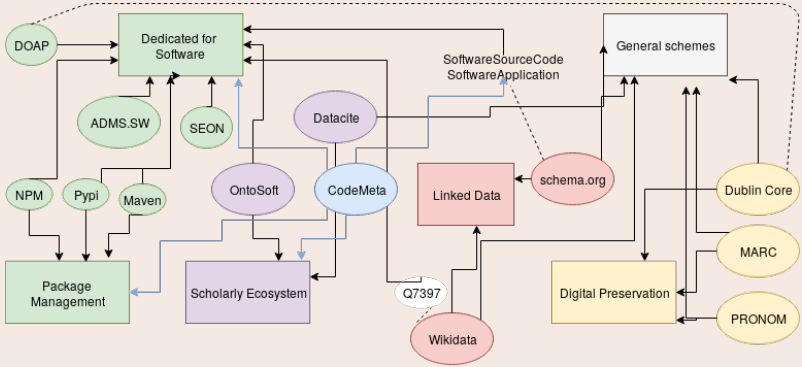
Later on

- Support embargo/access control

Outline

Collecting metadata for 60+ million projects

Landscape of Software Ontologies

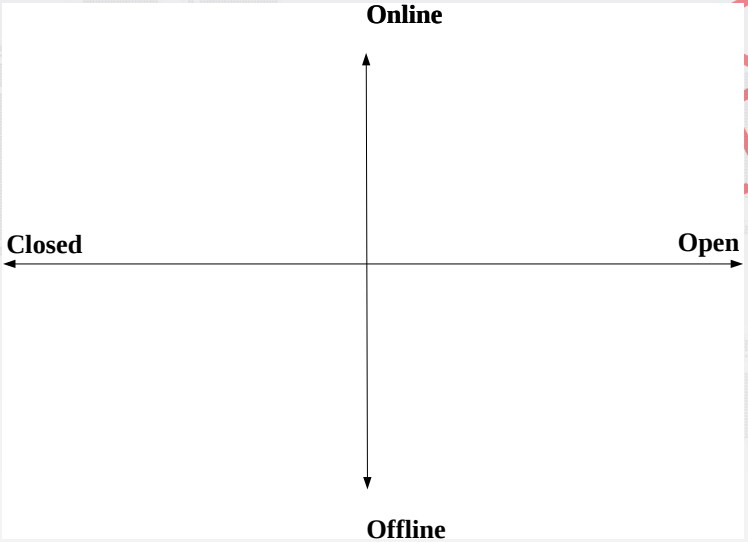


It's the real world!

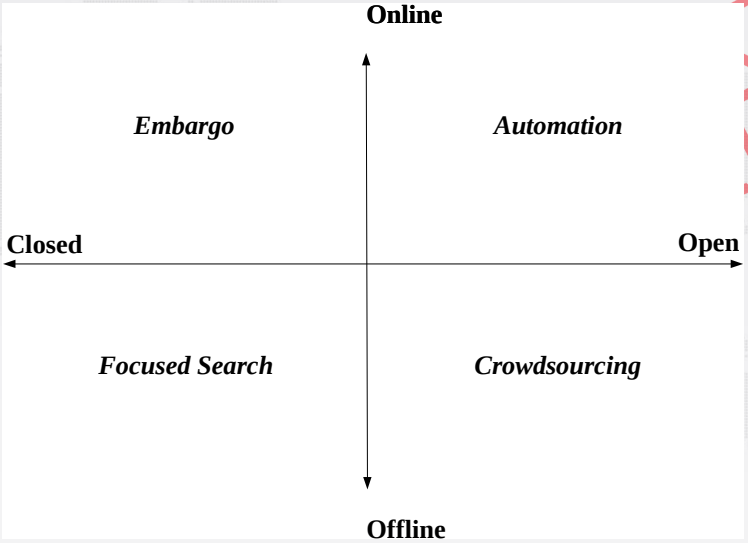
reconcile metadata from different origins, handle conflicts, synthesise missing information, classify (automatically) the projects, etc.

Outline

All the source code



All the source code, strategies



Online, open source code: automation overview