Software Heritage
building a community to safeguard the Software Commons

Nicolas Dandrimont
Software Heritage

September 10th 2023
DebConf 23
1. Software and Source Code

2. Software Heritage: a mission at the service of Humankind

3. Diving deeper into our Features

4. Software Heritage Infrastructure

5. Concluding remarks
Software is all around us
Software is built from *Source Code*

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.”
Software is built from *Source Code*

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)

<table>
<thead>
<tr>
<th>P63SP0T3</th>
<th>CA BIT6</th>
<th># IS THE LR ANTENNA IN POSITION 1 YET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EXTEND</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RND</td>
<td>CHAN33</td>
</tr>
<tr>
<td></td>
<td>EXTEND</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BZF</td>
<td>P63SP0T4 # BRANCH IF ANTENNA ALREADY IN POSITION 1</td>
</tr>
<tr>
<td></td>
<td>CAF</td>
<td>CODE508 # ASTRONAUT: PLEASE CRANK THE</td>
</tr>
<tr>
<td></td>
<td>TC</td>
<td>BANKCALL # SILLY THING AROUND</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>GOPERF1</td>
</tr>
<tr>
<td></td>
<td>TCF</td>
<td>GOTOP00H # TERMINATE</td>
</tr>
<tr>
<td></td>
<td>TCF</td>
<td>P63SP0T3 # PROCEED SEE IF HE’S LYING</td>
</tr>
<tr>
<td>P63SP0T4</td>
<td>TC</td>
<td>BANKCALL # ENTER INITIALIZING LANDING RADAR</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>SETPOS1</td>
</tr>
<tr>
<td></td>
<td>TC</td>
<td>POSTJUMP # OFF TO SEE THE WIZARD ...</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>BURNBABY</td>
</tr>
</tbody>
</table>
Software is built from **Source Code**

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)**

```
P63SP0T3  CA BIT7  # IS THE LR ANTENNA IN POSITION 1 YET
          EXTEND
          RAND  CHAN33
          EXTEND
          BZF  P63SP0T4  # BRANCH IF ANTENNA ALREADY IN POSITION 1
          CAF  CODE50B
          TC  BANKCALL
          CADR  GOPERF1
          TCF  GOT0PO01
          TCF  P63SP0T3  # PROCEED SEE IF HE'S LYING
          P63SP0T4
          TC  BANKCALL
          CADR  SETPO01
          TC  POSTJUMP
          CADR  BURNBABY
          # OFF TO SEE THE WIZARD ...
```

**Quake III source code (excerpt)**

```
float Q_rsqr(t(float number)
{
    long i;
    float x2, y;
    const float threehalves = 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 * (threehalves - (x2 * y + y)); // 1st iteration
    // y = y * (threehalves - (x2 * y + y)); // 2nd iteration, this
    // can be removed
    return y;
}
```
Software is built from **Source Code**

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)**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>P83SPOT3</td>
<td>CA BIT6</td>
<td># IS THE LR ANTENNA IN POSITION 1 YET</td>
</tr>
<tr>
<td></td>
<td>EXTEND</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAND</td>
<td>CHAN33</td>
</tr>
<tr>
<td></td>
<td>EXTEND</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BZF</td>
<td>P83SPOT4</td>
</tr>
<tr>
<td></td>
<td>CAF</td>
<td>CODE508</td>
</tr>
<tr>
<td></td>
<td>TC</td>
<td>BANKCALL</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>GOPERF1</td>
</tr>
<tr>
<td></td>
<td>TCF</td>
<td>GOTOP000H</td>
</tr>
<tr>
<td></td>
<td>TCF</td>
<td>P63SPOT3</td>
</tr>
<tr>
<td>P63SPOT4</td>
<td>TC</td>
<td>BANKCALL</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>SETPOS1</td>
</tr>
<tr>
<td></td>
<td>TC</td>
<td>POSTJUMP</td>
</tr>
<tr>
<td></td>
<td>CADR</td>
<td>BURNBABY</td>
</tr>
</tbody>
</table>

**Quake III source code (excerpt)**

```c
float Q_rsqrt (float number) {
    long i;
    float x2, y;
    const float threehalves = 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 * ( threehalves - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalves - ( x2 * y * y ) ); // 2nd iteration, this
    // can be removed

    return y;
}
```

Len Shustek, Computer History Museum 2006

“Source code provides a view into the mind of the designer.”
Experts call for greater recognition of software source code as heritage for sustainable development

6 November 2018

UNESCO, Inria, Software Heritage invite
40 international experts meet in Paris …
Experts call for greater recognition of software source code as heritage for sustainable development

6 November 2018

UNESCO, Inria, Software Heritage invite 40 international experts meet in Paris …

The call is published on February 2019
UNESCO, Inria, Software Heritage invite 40 international experts meet in Paris …

“The call is published on February 2019

“Recognise software source code as a fundamental enabler in all aspects of human endeavour”
Yuval Noah Harari (on COVID 19)

“The real antidote [to the pandemic] is scientific knowledge and global cooperation.”
Yuval Noah Harari (on COVID 19)

“The real antidote [to the pandemic] is scientific knowledge and global cooperation.”

Software powers modern scientific research

The top 100 papers

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

Nature, October 2014
Yuval Noah Harari (on COVID 19)

“The real antidote [to the pandemic] is scientific knowledge and global cooperation.”

Software powers modern scientific research

The top 100 papers

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

Nature, October 2014

We can still talk to the early inventors

"Telling historical stories is the best way to teach. It’s much easier to understand something if you know the threads it is connected to."

Donald E. Knuth
Len Shustek
CACM, January 2021
Yuval Noah Harari (on COVID 19)

“The real antidote [to the pandemic] is scientific knowledge and global cooperation.”

Software powers modern scientific research

The top 100 papers

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

Nature, October 2014

We can still talk to the early inventors

"Telling historical stories is the best way to teach. It’s much easier to understand something if you know the threads it is connected to."

Donald E. Knuth
Len Shustek
CACM, January 2021

We need a dedicated infrastructure to preserve and share all this knowledge!
Enhancing software Reuse, Security and Transparency

Software complexity is growing... it is important to Know Your SoftWare (KYSW)
Enhancing software Reuse, Security and Transparency

Software complexity is growing... it is important to Know Your SoftWare (KYSW)

Regulation on Software Updates
Recording [...] software versions relevant to a vehicle type

UN Regulations on Cybersecurity, June 2020
Enhancing software Reuse, Security and Transparency

Software complexity is growing... it is important to Know Your SoftWare (KYSW)

Regulation on Software Updates
Recording [...] software versions relevant to a vehicle type
UN Regulations on Cybersecurity, June 2020

Politique publique de la donnée, des algorithmes et des codes sources
…animer les écosystèmes des…réutilisateurs du source code
Circulaire du Premier Ministre, 27 Avril 2021, France
Enhancing software Reuse, Security and Transparency

Software complexity is growing... it is important to Know Your SoftWare (KYSW)

Regulation on Software Updates
Recording [...] software versions relevant to a vehicle type

UN Regulations on Cybersecurity, June 2020

Politique publique de la donnée, des algorithmes et des codes sources

...animer les ecosystèmes des...réutilisateurs du source code

Circulaire du Premier Ministre, 27 Avril 2021, France

Sec. 4. Enhancing Software Supply Chain Security

ensuring and attesting, to the extent practicable, to the integrity and provenance of open source software

May 2021 POTUS Executive Order
Enhancing software Reuse, Security and Transparency

Software complexity is growing... it is important to Know Your SoftWare (KYSW)

Regulation on Software Updates
Recording [...] software versions relevant to a vehicle type
UN Regulations on Cybersecurity, June 2020

Politique publique de la donnée, des algorithmes et des codes sources
...animer les écosystèmes des...réutilisateurs du source code
Circulaire du Premier Ministre, 27 Avril 2021, France

Sec. 4. Enhancing Software Supply Chain Security
ensuring and attesting, to the extent practicable, to the integrity and provenance of open source software
May 2021 POTUS Executive Order

We need a trusted knowledge base with software provenance!
Software source code is fragile

**Endangered source code …**

- *link rot*: projects are created, moved around, removed
- *data rot*: physical media with legacy software decay
- *platform consolidation* endangers repositories
  - 2015 Google Code and Gitorious.org shutdown: ~1M
  - 2019 Bitbucket mercurial phase out: ~250,000
  - 2022 GitLab.com: remove inactive projects?

---

Broken links and missing pieces in the web of knowledge of humankind

Bottomline: we need a global, long term effort to build a universal archive of all software source code and make it sustainable.
Software source code is fragile

Endangered source code …

- **link rot**: projects are created, moved around, removed
- **data rot**: physical media with legacy software decay
- **platform consolidation** endangers repositories
  - 2015 Google Code and Gitorious.org shutdown: ~1M
  - 2019 Bitbucket mercurial phase out: ~250,000
  - 2022 GitLab.com: remove inactive projects?

… is endangered knowledge!

broken links and missing pieces in the *web of knowledge* of humankind
Software source code is fragile

Endangered source code …

- link rot: projects are created, moved around, removed
- data rot: physical media with legacy software decay
- platform consolidation endangers repositories
  - 2015 Google Code and Gitorious.org shutdown: ~1M
  - 2019 Bitbucket mercurial phase out: ~250,000
  - 2022 GitLab.com: remove inactive projects?

… is endangered knowledge!

broken links and missing pieces in the web of knowledge of humankind

Bottomline: we need a global, long term effort

... to build a universal archive of all software source code
... and make it sustainable
Outline

1. Software and Source Code
2. Software Heritage: a mission at the service of Humankind
3. Diving deeper into our Features
4. Software Heritage Infrastructure
5. Concluding remarks
Software Heritage in a nutshell

**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
Unveiled in 2016

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
Unveiled in 2016

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
Software Heritage in a nutshell

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

Nicolas Dandrimont olasd@softwareheritage.org @olasd@mastodon.opportunis.me (CC-BY 4.0)
Today: a universal software archive, as a shared infrastructure

One infrastructure open and shared
Today: a *universal* software archive, as a shared infrastructure

One infrastructure
open and shared

Software Heritage

The largest archive ever built
Today: a *universal* software archive, as a shared infrastructure

One infrastructure open and shared

The largest archive ever built
An operational, evolving infrastructure

Harvest and archive

Reference (25 billion SWHIDs)
Intrinsic, decentralised, cryptographically strong identifiers
Now in SPDX 2.2, Wikidata, ISO is coming

Nicolas Dandrimont olasd@softwareheritage.org @olasd@mastodon.opportunis.me (CC-BY 4.0)
An operational, evolving infrastructure

Harvest and archive

- Forges
- Distros
- Package repos

Software Heritage Archive
- Merkle DAG + blob storage
- Loading & deduplication

- GitHub lister
- GitLab lister
- Debian lister
- PyPi lister

- Git loader
- Mercurial loader
- Debian source package loader
- PyPi source package loader

- Listing (full/incremental)
- Scheduling

- tar
- origins
- snapshots
- releases
- revisions
- directories
- contents
Harvest and archive

- save.softwareheritage.org
- deposit.softwareheritage.org
An operational, evolving infrastructure

Harvest and archive

- save.softwareheritage.org
- deposit.softwareheritage.org

Reference (25 billion SWHIDs)

Intrinsic, decentralised, cryptographically strong identifiers

Now in SPDX 2.2, Wikidata, ISO is coming
Growing adoption

Adoption in Open Science
reference archive (example) for research software

Adoption in Industry and Public Administration
reference archive and knowledge base for open source software
Growing adoption

Adoption in Open Science
reference archive (example)
for research software

Adoption in Industry and Public Administration
reference archive and knowledge base
for open source software
A revolutionary infrastructure

The graph of public software development

All software development in a single graph …

- enable traceability
A revolutionary infrastructure

The graph of public software development

All software development in a single graph ...

- enable traceability

The global ledger of public code

... a Merkle graph

- ensure integrity

Nicolas Dandrimont  olasd@softwareheritage.org  @olasd@mastodon.opportunis.me  (CC-BY 4.0)
A revolutionary infrastructure

The **graph** of public software development

All software development in a single graph ...

- enable traceability

---

The **global ledger** of public code

... a Merkle graph

- ensure integrity

---

A **pillar** of Open Science

Reference **archive** of Research Software

- reproducibility
- reference
A revolutionary infrastructure

The *graph* of public software development

All software development in a single graph …

- enable traceability

The *global ledger* of public code

… a Merkle graph

- ensure integrity

A *pillar* of Open Science

Reference *archive* of Research Software

- reproducibility
- reference

Reference platform for *Big Code*

uniform data structure

- large scale studies
- machine learning, AI, …
A walkthrough

General
- Browse the archive, get and use SWHIDs, e.g. Apollo 11 excerpt, Quake III excerpt
- Trigger archival in one click with the browser extension

Open Science
- Curated deposit via HAL, e.g.: LinBox, SLALOM, Givaro, SumGra, Coq proof, …
- Cite software with the biblatex-software style, e.g.: article from IPOL

History of software: rescuing landmark legacy software
see SWHAP process, Software Stories, and SWHAP Days 2022

Public code
Archived source code from code.gouv.fr
An international, non profit initiative for the long term

Sharing the vision

And many more ...

www.softwareheritage.org/support/testimonials
An international, non profit initiative for the long term

Sharing the vision

Donors, members, sponsors

And many more ...

www.softwareheritage.org/support/testimonials

Diamond sponsor

Platinum sponsors

Gold sponsors

Silver sponsors

Bronze sponsors

Nicolas Dandrimont olasd@softwareheritage.org @olasd@mastodon.opportunis.me (CC-BY 4.0)
Long-term storage

Object Storage

Challenge
- Storage of and efficient access to the individual versions of archived source code files
- 15 billion objects, median size of 3 kB

Implementation
- multiple backends implementing a common interface (fs, Ceph, public clouds)
- object packing on regular distributed block storage
- maintenance of 3 copies stored at different locations on different backends
Long-term storage

Object Storage

Challenge
- Storage of and efficient access to the individual versions of archived source code files
- 15 billion objects, median size of 3 kB

Implementation
- multiple backends implementing a common interface (fs, Ceph, public clouds)
- object packing on regular distributed block storage
- maintenance of 3 copies stored at different locations on different backends

Graph storage

Challenge
- storing the global history of software development
- resilient storage & efficient traversals with $>10^{10}$ vertices, $>10^{11}$ edges

Implementation
- store the vertices and edges for resilience in a "simple" Key-Value store (backed by Cassandra or PostgreSQL)
- compressed in-memory snapshots of the graph for traversals
Mirroring

Multiple challenges to the resilience of the infrastructure

- intentional or unintentional destruction
- legal framework changes
- permanence of the umbrella organization

Building a mirror network

- Avoid single point of (organizational) failure
- Host content under different jurisdictions
- Current deployments:
  - ENEA (Italy)
  - GRNet for EOSC (Greece)
Archival framework

**Listers**

Listing the contents of hosting platforms to make their contents available for archival

- VCS forges (GitHub, GitLab, Gitea, Forgejo, pagure, …)
- Distributions (Debian-based, Red Hat-based, language ecosystems like PyPI, Rubygems, …)
- **Documentation of swh.lister**
Archival framework

Listers
Listing the contents of hosting platforms to make their contents available for archival
- VCS forges (GitHub, GitLab, Gitea, Forgejo, pagure, …)
- Distributions (Debian-based, Red Hat-based, language ecosystems like PyPI, Rubygems, …)
- Documentation of swh.lister

Loaders
Converting source code, as published, into our common data model and ingesting the data
- VCS (git, Subversion, CVS, bazaar, mercurial), with full development history
- Software releases (as tarballs or individual files) from other distribution platforms
- Archive coverage
Save code now

- https://save.softwareheritage.org/
- Separate, autoscaling infrastructure
- Support for all available VCSes
On-demand archival

Save code now

- https://save.softwareheritage.org/
- Separate, autoscaling infrastructure
- Support for all available VCSes

Save Code Now APIs

- public access: Documentation of API
- raised rate-limits available to partners
- Browser extension
Curated deposit of software

**Push-based architecture**

- Based on the interoperable SWORD archival protocol
- Push a set of tarballs, receive a pointer to the SWHID of the archived software
- [Documentation of swh.deposit](#)
Curated deposit of software

Push-based architecture
- Based on the interoperable SWORD archival protocol
- Push a set of tarballs, receive a pointer to the SWHID of the archived software
- Documentation of swh.deposit

Audience
- Academic partners: automatic ingestion of source code deposited alongside open access research papers
- Industry partners: deposit of the complete corresponding source code for GPLv3 compliance in products
Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”
Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”
The SWHAP process, with UNESCO and University of Pisa

Paris Call on Software Source Code

“We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

- **Rescue** Legacy Software from different media
  - physical
  - digital
    - legacy / unsupported
    - recent / supported
The SWHAP process, with UNESCO and University of Pisa

Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

- **Rescue** Legacy Software from different media
  - physical
  - digital
  - legacy / unsupported
  - recent / supported

- **Curate** the code
  - reconstructing the development history
  - collecting metadata
The SWHAP process, with UNESCO and University of Pisa

Paris Call on Software Source Code

“We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

- **Rescue** Legacy Software from different media
  - physical
  - digital
  - legacy / unsupported
  - recent / supported

- **Curate** the code
  - reconstructing the development history
  - collecting metadata

- And **illustrate** with dedicated presentations
Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”
“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”
Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

- **Expand** the SWHAP scope to:
  - documents
  - media (videos, pictures, images, etc.)
  - oral history

---

The Pisa Collection

Stories of landmark legacy code (Beta Version)

3 stories in this collection

Search Collection...

TAUmus
Paris Call on Software Source Code

“We call to support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive.”

- Expand the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history

- Preserve and Present all this material
Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

- Expand the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history

- Preserve and Present all this material

- Share process and tools (all open source!)
  - with museums, archives and all interested parties
Paris Call on Software Source Code

“We call to support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

- Expand the SWHAP scope to documents, media (videos, pictures, images, etc.), oral history
- Preserve and Present all this material
- Share process and tools (all open source!) with museums, archives and all interested parties

see this live on the Software Stories website, and get the guide

Nicolas Dandrimont olasd@softwareheritage.org @olasd@mastodon.opportunis.me (CC-BY 4.0)
Add Forge now

**Challenge**
discover the wealth of smaller scale hosting platforms based on GitLab, Forgejo, Gitea, Gogs and other self-hostable forge software

**Implementation**
Submission form on the main archive website for users to submit new forges they’ve discovered, workflow for ingestion

**Documentation of Add forge now**
**Add Forge now**

**Challenge**

discover the wealth of smaller scale hosting platforms based on GitLab, Forgejo, Gitea, Gogs and other self-hostable forge software

**Implementation**

Submission form on the main archive website for users to submit new forges they’ve discovered, workflow for ingestion

**Documentation of Add forge now**

**Automation**

- dedicated infrastructure for sanity checking the submission, performing the first listing and loading of all the contents
- configurable concurrency (to avoid overloading smaller platforms)
- Work in progress automation of the full process through GitLab pipelines
Large scale analysis

Graph compression pipeline

Challenge

efficient queries on a graph that has hundreds of billions of edges?

Implementation

- Compress it and hold that representation in memory!
- From tens of terabytes of raw data, to a data structure that can be held in a few hundred GB of RAM
- Documentation of swh.graph
Large scale analysis

Graph compression pipeline

Challenge
efficient queries on a graph that has hundreds of billions of edges?

Implementation
- Compress it and hold that representation in memory!
- From tens of terabytes of raw data, to a data structure that can be held in a few hundred GB of RAM
- Documentation of swh.graph

Dataset exports
- Run your own queries on the full graph of software development
- Available for download, or for direct use on Amazon Athena
- Documentation of swh.dataset
- Ethical charter for bulk access
From humble beginnings
To a very large scale deployment
A very large scale deployment

Our main storage requirements in a nutshell

- More than 1PB of source code files (replicated 3 times by Software Heritage)
- More than 100 TB used for (resilient) storage of the graph
- Infrastructure support for mirroring: 100 TB kafka deployment (~30TB of data used)
A very large scale deployment

Our main storage requirements in a nutshell

- More than 1PB of source code files (replicated 3 times by Software Heritage)
- More than 100 TB used for (resilient) storage of the graph
- Infrastructure support for mirroring: 100 TB kafka deployment (~30TB of data used)

Infrastructure components

- on-site:
  - 4 racks of servers, all running Debian
    - proxmox cluster
    - kubernetes clusters (bare metal and VMs)
    - special purpose clusters (PostgreSQL, kafka, Cassandra, elasticsearch, …)
  - 1 rack of network equipment

- off-site:
  - Ceph objstorage cluster: 2 racks
  - Azure resources (blob storage, bare VMs, AKS clusters)
  - AWS resources (S3 for objstorage and dataset exports, Athena for queries)
A very large scale deployment

Our main storage requirements in a nutshell

- More than 1PB of source code files (replicated 3 times by Software Heritage)
- More than 100 TB used for (resilient) storage of the graph
- Infrastructure support for mirroring: 100 TB kafka deployment (~30TB of data used)

Infrastructure components

**on-site:**
- 4 racks of servers, all running Debian
- proxmox cluster
- kubernetes clusters (bare metal and VMs)
- special purpose clusters (PostgreSQL, kafka, Cassandra, elasticsearch, …)
- 1 rack of network equipment

**off-site:**
- Ceph objstorage cluster: 2 racks
- Azure resources (blob storage, bare VMs, AKS clusters)
- AWS resources (S3 for objstorage and dataset exports, Athena for queries)
Automation

Low-level deployments

- Everything running Debian, of course
- terraform for provisioning **virtual machines** and **cloud resources**
- puppet for deployment of OS components
Automation

Low-level deployments
- Everything running Debian, of course
- `terraform` for provisioning virtual machines and cloud resources
- `puppet` for deployment of OS components

Kubernetes
- Using `Rancher` and `rke2` as K8s distribution
- `ArgoCD` for continuous deployment (configuration)
- `Helm charts` for deploying the Software Heritage stack
- `Jenkins pipelines` for image building and Helm chart updates
- Images pushed to the GitLab container registry
Team organization

A regular Free Software project

- Organized in public
- Browse our GitLab CE instance on gitlab.softwareheritage.org
- Jenkins for CI automation (WIP: GitLab CI)
- real-time discussions on IRC bridged to matrix; open mailing lists
- Browse our documentation on docs.softwareheritage.org
Team organization

A regular Free Software project

- Organized in public
- Browse our GitLab CE instance on gitlab.softwareheritage.org
- Jenkins for CI automation (WIP: GitLab CI)
- real-time discussions on IRC bridged to matrix; open mailing lists
- Browse our documentation on docs.softwareheritage.org

Infrastructure as code

- All our deployment manifests are published in the open as well
- Repository access info

Nicolas Dandrimont  olasd@softwareheritage.org  @olasd@mastodon.opportunis.me
A growing and active community

Team

Let us save what remains: . . . by such a multiplication of copies, as shall place them beyond the reach of accident.

— Thomas Jefferson

Enea, GRNET, . . . CEA, RedHat

Nicolas Dandrimont olasd@softwareheritage.org @olasd@mastodon.opportunis.me (CC-BY 4.0)

Software Heritage softwareheritage.org 2023-09-10 28 / 30
A growing and active community

Team

Ambassadors

“Let us save what remains: . . . by such a multiplication of copies, as shall place them beyond the reach of accident.” — Thomas Jefferson

Nicolas Dandrimont olasd@softwareheritage.org @olasd@mastodon.opportunis.me Software Heritage softwareheritage.org 2023-09-10 28 / 30
A growing and active community

Team

Ambassadors

Foundations and grantees

- Castalia, CottageLabs
- EasterEggs, OcamlPro
- Octobus, Sperling, Tweag.io
- DataCurrent

"Let us save what remains: . . . by such a multiplication of copies, as shall place them beyond the reach of accident."
— Thomas Jefferson

Nicolas Dandrimont olasd@softwareheritage.org @olasd@mastodon.opportunis.me

Software Heritage softwareheritage.org 2023-09-10
A growing and active community

Team

Foundations and grantees

- Castalia, CottageLabs
- EasterEggs, OcamlPro
- Octobus, Sperling, Tweag.io
- DataCurrent

Ambassadors

- Castalia, CottageLabs
- EasterEggs, OcamlPro
- Octobus, Sperling, Tweag.io
- DataCurrent

Mirrors and storage partners

“Let us save what remains: … by such a multiplication of copies, as shall place them beyond the reach of accident.”

— Thomas Jefferson

Enea, GRNET, …

CEA, RedHat

Nicolas Dandrimont olasd@softwareheritage.org @olasd@mastodon.opportunis.me

Software Heritage softwareheritage.org 2023-09-10
Outline

1. Software and Source Code
2. Software Heritage: a mission at the service of Humankind
3. Diving deeper into our Features
4. Software Heritage Infrastructure
5. Concluding remarks
A call to realize a grand vision

Bring together academia, industry, civil society and governments to build

"a global infrastructure for open and better software at the service of humankind"
A call to realize a grand vision

Bring together academia, industry, civil society and governments to build

"a global infrastructure for open and better software at the service of humankind"

Software Heritage is the first brick …

- vendor neutral, multi-stakeholder
- open source, non profit
- a worldwide initiative
- a long term initiative
A call to realize a grand vision

Bring together academia, industry, civil society and governments to build
"a global infrastructure for open and better software at the service of humankind"

Software Heritage is the first brick …
- vendor neutral, multi-stakeholder
- open source, non profit
- a worldwide initiative
- a long term initiative

… that will enable
- archival, reference, integrity
- qualification, sharing and reuse
- a global software knowledge base
- test and deploy world class tooling

You can help!
use, adopt, advocate, contribute, fund, support, join

Nicolas Dandrimont olasd@softwareheritage.org @olasd@mastodon.opportunis.me (CC-BY 4.0)
A call to realize a grand vision

Bring together academia, industry, civil society and governments to build

"a global infrastructure for open and better software at the service of humankind"

Software Heritage is the first brick …

- vendor neutral, multi-stakeholder
- open source, non profit
- a worldwide initiative
- a long term initiative

… that will enable

- archival, reference, integrity
- qualification, sharing and reuse
- a global software knowledge base
- test and deploy world class tooling

A lot more is needed

Software Heritage can be the catalyser of a way bigger undertaking
A call to realize a grand vision

Bring together academia, industry, civil society and governments to build
"a global infrastructure for open and better software at the service of humankind"

Software Heritage is the first brick …
- vendor neutral, multi-stakeholder
- open source, non profit
- a worldwide initiative
- a long term initiative

… that will enable
- archival, reference, integrity
- qualification, sharing and reuse
- a global software knowledge base
- test and deploy world class tooling

A lot more is needed
Software Heritage can be the catalyser of a way bigger undertaking

You can help!
use, adopt, advocate, contribute, fund, support, join
A call to realize a grand vision

Bring together academia, industry, civil society and governments to build

"a global infrastructure for open and better software at the service of humankind"

Software Heritage

www.softwareheritage.org @swheritage

The Library of Alexandria of code

- recover the past
- structure the future
- rebuild trust in science

The Very Large Telescope for Source code

- explore and reuse
- better, more secure software
  for society as a whole