Software Heritage

Peeking under the hood of the largest archive of Source Code in the world

Nicolas Dandrimont
Tech Lead, Operations team
nicolas.dandrimont@inria.fr

September 29, 2026



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

1. Software as a pillar of modern society
2. Meet Software Heritage: open, non profit, multistakeholder
3. What's under the hood?
4. Tech team organization
5. Tech stack
6. Team's "Continuous Improvement"
7. Conclusions

Software is all around us

Invisible fabric of digital society



Knowledge is in the source

```
/**
 * @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=
           Global search and replace of 'unsigned short int' with 'int' would remove this limit, but use more memo
        */

        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) with
        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;
};
```

Covid Sim (excerpt)

Software is all around us

Invisible fabric of digital society



Knowledge is in the source

```
/**
 * @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=
           Global search and replace of 'unsigned short int' with 'int' would remove this limit, but use more memo
        */

        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) with
        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;
};
```

Covid Sim (excerpt)

Len Shustek, Computer History Museum 2006

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

A Global Undertaking

From all continents

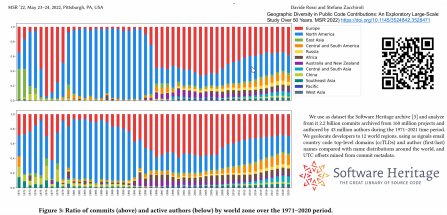
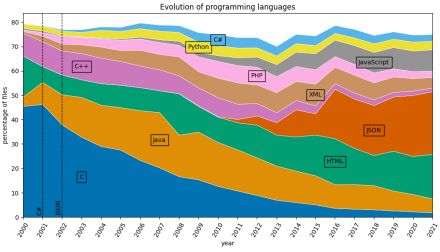
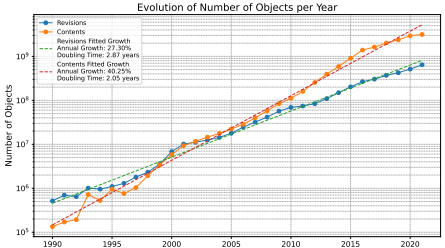


Figure 3: Ratio of commits (above) and active authors (below) by world zone over the 1971–2020 period.

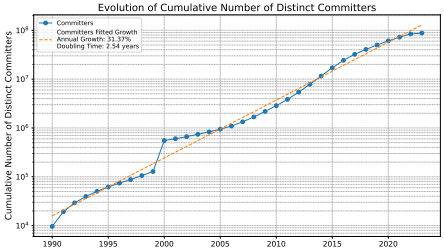
Many programming languages



An exponential growth (code)

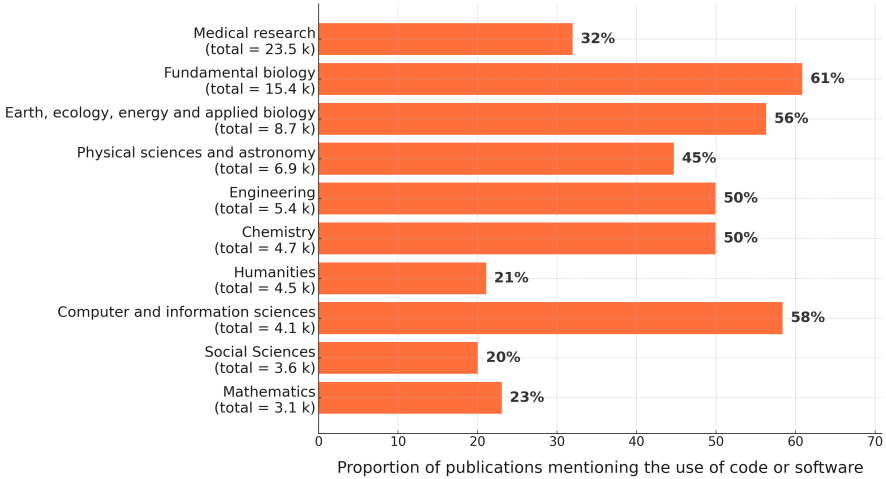


An exponential growth (contributors)



A pillar of Open Science

Pillar of Science across all research areas



Source code is special

Software *evolves* over time

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

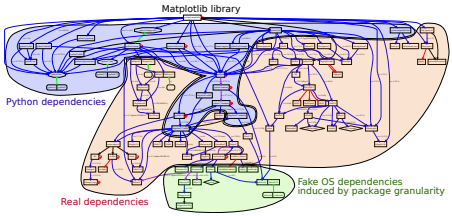
Source code is special

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*



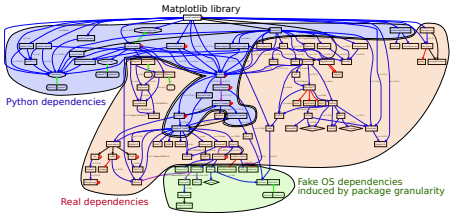
Source code is special

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 …

we need a dedicated infrastructure:

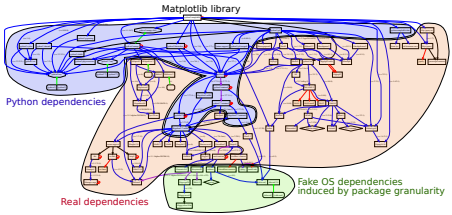
Source code is special

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 …

we need a dedicated infrastructure: now we have it!

Outline

1. Software as a pillar of modern society
2. Meet Software Heritage: open, non profit, multistakeholder
3. What's under the hood?
4. Tech team organization
5. Tech stack
6. Team's "Continuous Improvement"
7. Conclusions

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Inria with unesco



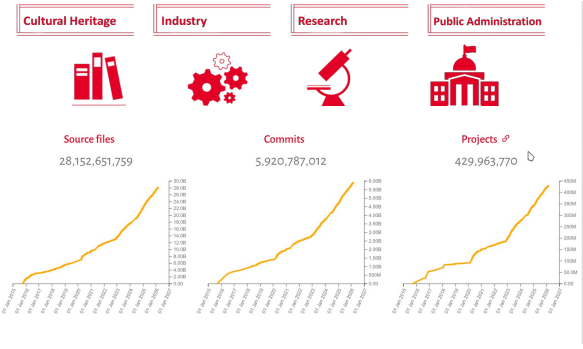
Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Inria with unesco



The largest open source code archive: one infrastructure, open, shared, non profit

Unique digital common good *built in France since 2015*



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Inria with unesco



The largest open source code archive: one infrastructure, open, shared, non profit

Unique digital common good *built in France since 2015*

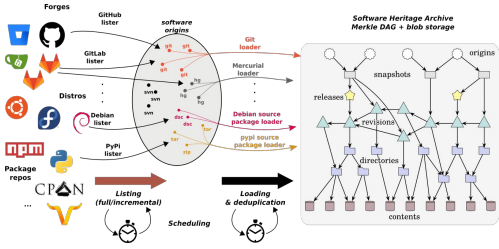
Cultural Heritage | **Industry** | **Research** | **Public Administration**

Source files
28,152,651,759

Commits
5,920,787,012

Projects
429,963,770





5000+ platforms

All versions, all history development in a single graph

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Inria with unesco



The largest open source code archive: one infrastructure, open, shared, non profit

Unique digital common good *built in France since 2015*

Cultural Heritage | **Industry** | **Research** | **Public Administration**

Source files
28,152,651,759

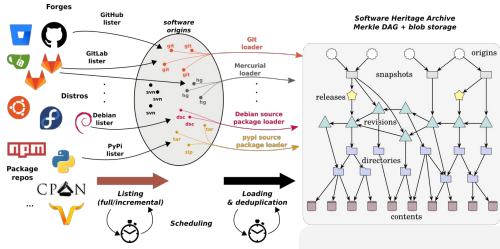


Commits
5,920,787,012



Projects
429,963,770





5000+ platforms

All versions, all history development in a single graph

- **50 × 10⁹** nodes
- **1000 × 10⁹** edges
- ~ **3 PB** of storage

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

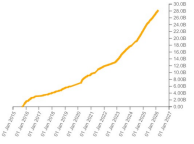
Inria with unesco

The largest open source code archive: one infrastructure, open, shared, non profit

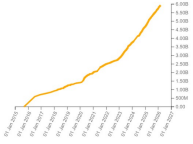
Unique digital common good *built in France since 2015*

Cultural Heritage | **Industry** | **Research** | **Public Administration**

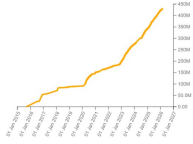
Source files
28,152,651,759

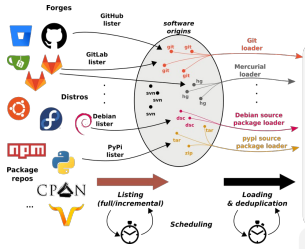


Commits
5,920,787,012



Projects
429,963,770





5000+ platforms

All versions, all history development in a single graph

- **50 × 10⁹** nodes
- **1000 × 10⁹** edges
~ **3 PB** of storage

A revolutionary **infrastructure** ensures **availability** guarantees **integrity** enables **traceability**

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

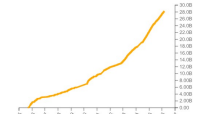
Inria with unesco

The largest open source code archive: one infrastructure, open, shared, non profit

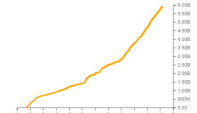
Unique digital common good *built in France since 2015*

Cultural Heritage | **Industry** | **Research** | **Public Administration**

Source files
28,152,651,759

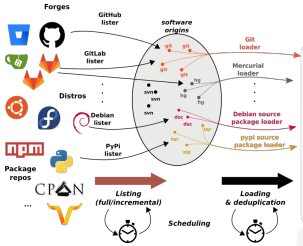


Commits
5,920,787,012



Projects
429,963,770





5000+ platforms

All versions, all history development in a single graph

- **50 × 10⁹** nodes
- **1000 × 10⁹** edges
- ~ **3 PB** of storage

A revolutionary **infrastructure** ensures **availability** guarantees **integrity** enables **traceability**

IBM Microsoft TII Technology Innovation Institute HUAWEI cnrs Infosys cea Ministère de l'Enseignement Supérieur et de la Recherche aws Google Université Paris Cité openinventionnetwork

AdaCore GitHub Université di Pisa Sorbonne Université Université de Lorraine Sant'Anna NWO université PARIS-SACLAY

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

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

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

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

Sharing the vision



And many more ...

www.softwareheritage.org/support/testimonials

Sharing the vision



And many more ...
www.softwareheritage.org/support/testimonials

Members and sponsors

Inria

Diamond sponsors: Microsoft, Technology Innovation Institute

Platinum sponsors: cnrs, Ministère de l'Enseignement supérieur et de la Recherche, Infosys, HUAWEI, CEA

Gold sponsors: aws, Google, GitHub, openinventionnetwork, Université Paris Cité

Silver sponsors: AdaCore, Ministère de l'Europe et des Affaires étrangères, Sorbonne Université, Università di Pisa

Bronze sponsors: Sant'Anna, Scuola Normale Superiore, Université de Lorraine, université PARIS-SACLAY

academia, industry, public sector, society
we are all concerned

A *universal* software archive, as a shared infrastructure

One infrastructure
open and shared



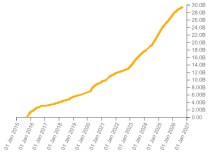
A *universal* software archive, as a shared infrastructure

One infrastructure
open and shared

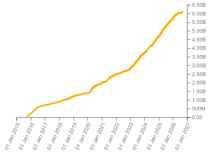


The largest archive ever built

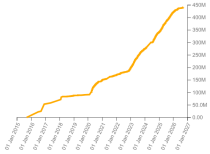
Source files
29,523,977,803



Commits
6,112,189,339



Projects
441,196,029



Directories
23,007,987,747

Authors
107,736,452

Releases
184,798,617

A *universal* software archive, as a shared infrastructure

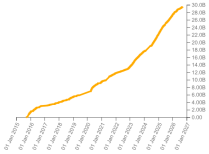
One infrastructure
open and shared



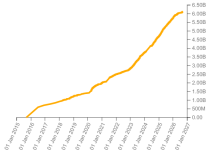
Inria unesco

The largest archive ever built

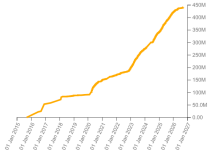
Source files
29,523,977,803



Commits
6,112,189,339



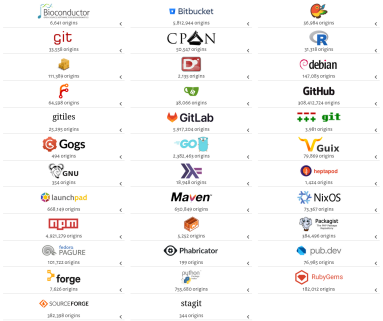
Projects
441,196,029



Directories
23,007,987,747

Authors
107,736,452

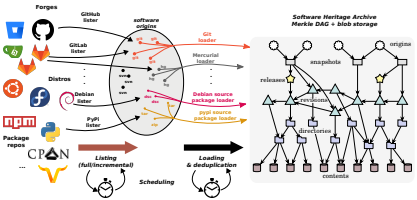
Releases
184,798,617



figures as of January 8 2026

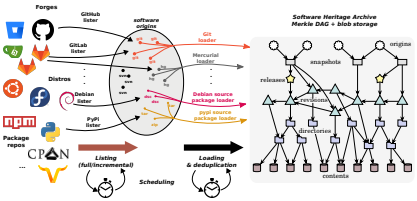
Addressing key needs in (Open) Science

Archive (28B+ files, 430M+ projects)



Addressing key needs in (Open) Science

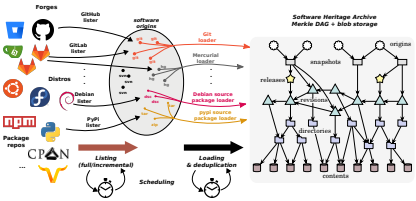
Archive (28B+ files, 430M+ projects)



- save now, updateswh, webhooks
- deposit.softwareheritage.org

Addressing key needs in (Open) Science

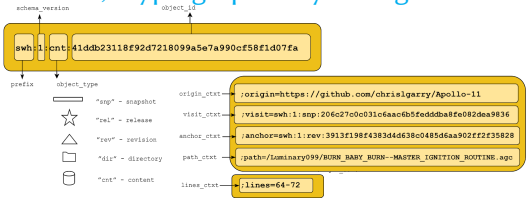
Archive (28B+ files, 430M+ projects)



- save now, updateswh, webhooks
- deposit.softwareheritage.org

Reference (50 billion SWHIDs)

Intrinsic, cryptographically strong IDs

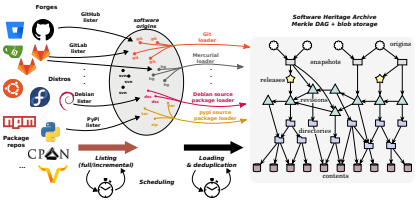


Now in SPDX 2.2, Wikidata

`https://swhid.org` - ISO/IEC 18670

Addressing key needs in (Open) Science

Archive (28B+ files, 430M+ projects)



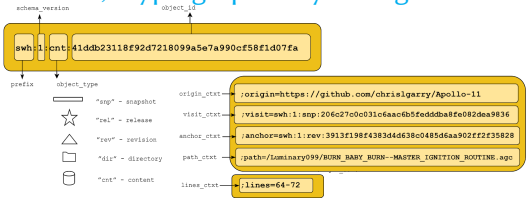
- save now, updateswh, webhooks
- deposit.softwareheritage.org

Describe

- *Intrinsic metadata* from source code
- Contributed the Codemeta generator

Reference (50 billion SWHIDs)

Intrinsic, cryptographically strong IDs

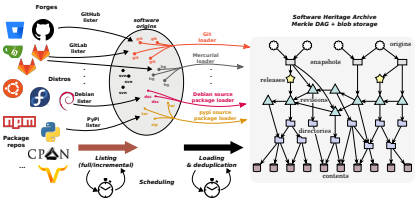


Now in SPDX 2.2, Wikidata

`https://swhid.org` - ISO/IEC 18670

Addressing key needs in (Open) Science

Archive (28B+ files, 430M+ projects)



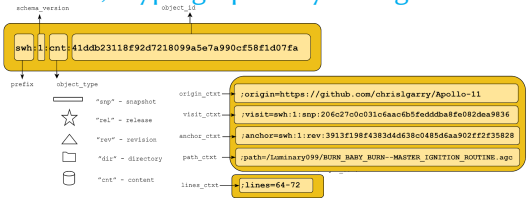
- save now, updateswh, webhooks
- deposit.softwareheritage.org

Describe

- *Intrinsic metadata* from source code
- Contributed the Codemeta generator

Reference (50 billion SWHIDs)

Intrinsic, cryptographically strong IDs



Now in SPDX 2.2, Wikidata

`https://swhid.org` - ISO/IEC 18670

Cite/Credit

- Contributed biblatex-software style
- Software Citation from the archive!

An example is worth a thousand words

- Browse + Reference [ISO 18670] (Apollo 11 [excerpt], your work may be already there !)
- Trigger archival, use the updateswh browser extension, configure the webhooks
- Cite from the archive with biblatex-software (CTAN, ACMART)
- Describe with Codemeta (use codemeta generator)
- Curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …
- Extracting all the software products for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools
- Example with Parmap: devel on Github, archive in SWH, curated deposit in HAL
- Example research articles:
  - compare Fig. 1 and conclusions in the 2012 version and the updated version
  - SWHID in a replication experiment

Outline

1. Software as a pillar of modern society
2. Meet Software Heritage: open, non profit, multistakeholder
3. What's under the hood?
4. Tech team organization
5. Tech stack
6. Team's "Continuous Improvement"
7. Conclusions

Infrastructure

- Inria Rocquencourt
  - 5 racks - 70 machines (Prod and Staging infrastructures)
  - Proxmox cluster (VMs hosting + networked storage for workloads)
  - Kubernetes clusters (Bare metal + VMs)
  - Bare metal data silos: Cassandra clusters; Kafka clusters; Elasticsearch clusters
  - Snowflake servers (graph compression, etc.)
- CEA Saclay
  - 3 racks - 42 machines (Ceph cluster - 6PB - for mass storage)

Infrastructure

- Inria Rocquencourt
  - 5 racks - 70 machines (Prod and Staging infrastructures)
  - Proxmox cluster (VMs hosting + networked storage for workloads)
  - Kubernetes clusters (Bare metal + VMs)
  - Bare metal data silos: Cassandra clusters; Kafka clusters; Elasticsearch clusters
  - Snowflake servers (graph compression, etc.)
- CEA Saclay
  - 3 racks - 42 machines (Ceph cluster - 6PB - for mass storage)
- Microsoft Azure
  - Object storage (copy of contents), hosted kubernetes clusters for ancillary infra
- Amazon S3
  - Object storage (copy of contents, dataset exports)

Core Team

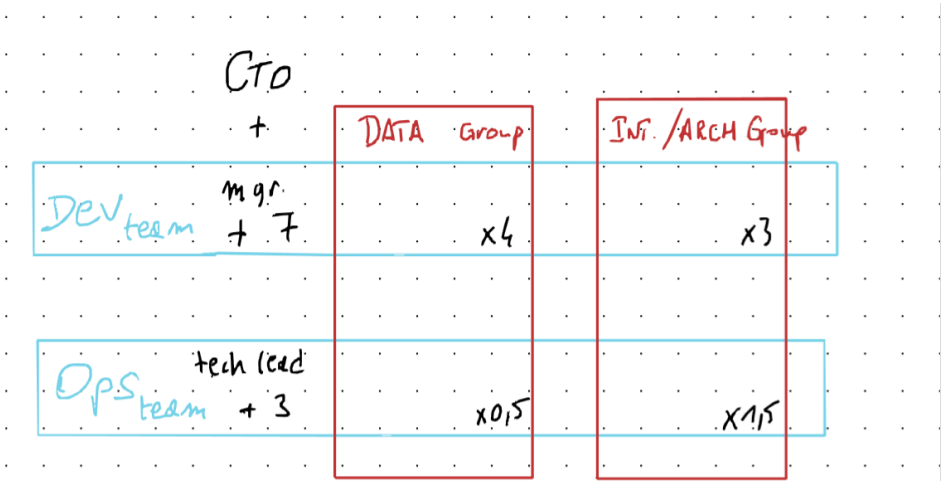


- 25 staff members (20 full time, 5 part time)
- 10 people in management / outreach / support
- 15 people in tech team (CTO + operations team + development team)
- 3 hires in progress

Outline

1. Software as a pillar of modern society
2. Meet Software Heritage: open, non profit, multistakeholder
3. What's under the hood?
4. Tech team organization
5. Tech stack
6. Team's "Continuous Improvement"
7. Conclusions

Tech team


CTO
+
DATA Group
INT. /ARCH Group
Dev team
mgr.
+ 7
x4
x3
Ops team
tech lead
+ 3
x0,5
x1,5

Day-to-day operations

Groups

- 2-week cycle for both groups (1h30 retrospective every 2 weeks)
- 15 mins "daily" checkpoints twice a week

Day-to-day operations

Groups

- 2-week cycle for both groups (1h30 retrospective every 2 weeks)
- 15 mins "daily" checkpoints twice a week
- Communication toolkit:
  - synchronous: jitsi + matrix
  - asynchronous: GitLab issues / Epics; mailing lists
  - collaborative editing: HedgeDoc

Day-to-day operations

Groups

- 2-week cycle for both groups (1h30 retrospective every 2 weeks)
- 15 mins "daily" checkpoints twice a week
- Communication toolkit:
  - synchronous: jitsi + matrix
  - asynchronous: GitLab issues / Epics; mailing lists
  - collaborative editing: HedgeDoc

Teams

- Weekly checkpoint for the dev & ops teams

Medium-term organization

Whole team

- Yearly (rolling) roadmap with high level priorities
- Monthly all hands meeting (management + outreach + tech team)

Medium-term organization

Whole team

- Yearly (rolling) roadmap with high level priorities
- Monthly all hands meeting (management + outreach + tech team)

Tech team

- Meeting every quarter (incl. retrospective and priorities for the next quarter)
- Regular individual meetings with CTO or team lead
- Regular (2-3 weekly) meeting for tech team leadership

Medium-term organization

Whole team

- Yearly (rolling) roadmap with high level priorities
- Monthly all hands meeting (management + outreach + tech team)

Tech team

- Meeting every quarter (incl. retrospective and priorities for the next quarter)
- Regular individual meetings with CTO or team lead
- Regular (2-3 weekly) meeting for tech team leadership

Management

- Bi-weekly management meetings (Management team + Tech, Outreach, Comms team leaders)
- Follow up on external projects (e.g. CodeCommons projects)

Outline

1. Software as a pillar of modern society
2. Meet Software Heritage: open, non profit, multistakeholder
3. What's under the hood?
4. Tech team organization
5. Tech stack
6. Team's "Continuous Improvement"
7. Conclusions

The Software Heritage tech stack

- Everything FLOSS (A)GPLv3+, developed in the open

The Software Heritage tech stack

- Everything FLOSS (A)GPLv3+, developed in the open
- Internal stack developed in house (Python + Rust) by the dev team

The Software Heritage tech stack

- Everything FLOSS (A)GPLv3+, developed in the open
- Internal stack developed in house (Python + Rust) by the dev team
- Deployment manifests mostly written by the ops team currently, with some handover to the devs in progress
  - terraform for low-level provisioning
  - puppet for system-level provisioning
  - Helm for kubernetes deployments

Continous Integration

- Unit testing of each Python / Rust module on Jenkins
  - WIP: Gitlab CI integration for newer projects

Continous Integration

- Unit testing of each Python / Rust module on Jenkins
  - WIP: Gitlab CI integration for newer projects
- Integration tests of the developed stack
  - Docker Compose / testinfra
  - Doubles as local development stack

Continous Integration

- Unit testing of each Python / Rust module on Jenkins
  - WIP: Gitlab CI integration for newer projects
- Integration tests of the developed stack
  - Docker Compose / testinfra
  - Doubles as local development stack
- Jenkins builds docker containers on releases
  - Pushes them to GitLab container registry
  - Creates a merge request to bump container versions in Helm chart

Continuous Deployment

- ArgoCD to manage state of apps on kubernetes clusters

Continuous Deployment

- ArgoCD to manage state of apps on kubernetes clusters
- 3 static Kubernetes clusters
  - "next-version": automatic deployment of latest container versions
  - staging: small-scale integration testing
  - production

Continuous Deployment

- ArgoCD to manage state of apps on kubernetes clusters
- 3 static Kubernetes clusters
  - "next-version": automatic deployment of latest container versions
  - staging: small-scale integration testing
  - production
- WIP: autonomy for deployments in the two groups
  - Separate K8s namespaces
  - …?

Outline

1. Software as a pillar of modern society
2. Meet Software Heritage: open, non profit, multistakeholder
3. What's under the hood?
4. Tech team organization
5. Tech stack
6. Team's "Continuous Improvement"
7. Conclusions

Team Challenges

Tech debt

- Organic growth over 10 years; little time spent on iterative improvements of the whole stack
- Very complex stack, impossible to hold in one's head all at once
- Unexpected consequences of seemingly small changes

Team Challenges

Tech debt

- Organic growth over 10 years; little time spent on iterative improvements of the whole stack
- Very complex stack, impossible to hold in one's head all at once
- Unexpected consequences of seemingly small changes

Limited resources

- Very difficult to do production-level tests at the large scale of the Archive
- Difficult to iterate once a solution is in place, within limited resources
- Data/schema migrations take months to years

Team Challenges

Tech debt

- Organic growth over 10 years; little time spent on iterative improvements of the whole stack
- Very complex stack, impossible to hold in one's head all at once
- Unexpected consequences of seemingly small changes

Limited resources

- Very difficult to do production-level tests at the large scale of the Archive
- Difficult to iterate once a solution is in place, within limited resources
- Data/schema migrations take months to years

Communication, internal vs. external

- Free software-style open-by-default challenges, Especially 2nd language English
- Documentation often an afterthought

Team training

Over the past 5 years

- "Software Craftsmanship"
  - Agile methods / Pair/Mob programming / …

Team training

Over the past 5 years

- "Software Craftsmanship"
  - Agile methods / Pair/Mob programming / …
- ITIL Foundations
  - Common "language" for project/product/service driven management

Team training

Over the past 5 years

- "Software Craftsmanship"
  - Agile methods / Pair/Mob programming / …
- ITIL Foundations
  - Common "language" for project/product/service driven management
- Kubernetes basics
  - Wanting to reduce the gap between dev and ops stack, and hand over some of the deployment concerns

Process improvements

Completed

- Recruited a tech writer a year ago to help keep our documentation useful (both for the team and for our users)

Process improvements

Completed

- Recruited a tech writer a year ago to help keep our documentation useful (both for the team and for our users)
- Introduction of SWHEP process (similar to RFC / PEP / ADR) for impactful changes in the tech stack

Process improvements

Completed

- Recruited a tech writer a year ago to help keep our documentation useful (both for the team and for our users)
- Introduction of SWHEP process (similar to RFC / PEP / ADR) for impactful changes in the tech stack
- On a recent project (OSPO-Radar), recruited a UI/UX designer to work on wireframes for the product and support our dev team

Process improvements (cont'd)

Ongoing

- Tech debt reduction
  - Automated dependency updates with Mend Renovate
  - GitLab CI in some projects
  - Migration to GitLab Enterprise for project management features

Process improvements (cont'd)

Ongoing

- Tech debt reduction
  - Automated dependency updates with Mend Renovate
  - GitLab CI in some projects
  - Migration to GitLab Enterprise for project management features
- Sharing ownership of our software deployment pipeline

Process improvements (cont'd)

Ongoing

- Tech debt reduction
  - Automated dependency updates with Mend Renovate
  - GitLab CI in some projects
  - Migration to GitLab Enterprise for project management features
- Sharing ownership of our software deployment pipeline
- Many iterative changes over time to make our organization more fluid

Outline

1. Software as a pillar of modern society
2. Meet Software Heritage: open, non profit, multistakeholder
3. What's under the hood?
4. Tech team organization
5. Tech stack
6. Team's "Continuous Improvement"
7. Conclusions

Conclusions

- Still a long road ahead!

Conclusions

- Still a long road ahead!
- Maybe early in building some structures (e.g. the Team/Group matrix)

Conclusions

- Still a long road ahead!
- Maybe early in building some structures (e.g. the Team/Group matrix)
- But we're continuing to grow the tech team

Conclusions

- Still a long road ahead!
- Maybe early in building some structures (e.g. the Team/Group matrix)
- But we're continuing to grow the tech team
- Some incremental improvements have felt slow but the results are starting to show

Questions?