From Dependence to Resilience

Building Ecumenical Digital Sovereignty for Software

Roberto Di Cosmo

CEO, Software Heritage
Inria & Université Paris Cité

6 May 2026
Auditorium CSI Next, Torino
CSI Piemonte — Sovranità digitale

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Outline

1. Why software source code matters
2. The regulation / infrastructure gap
3. Ecumenical digital sovereignty
4. Software Heritage: the ecumenical infrastructure
5. From archive to strategic backbone
6. A Very Large Telescope to explore the software development galaxy
7. We (happen to) have the answer at arm's reach
8. A path forward, together

Beneath cloud, data and infrastructure

Digital sovereignty

Often framed in terms of *cloud*, *data*, or *infrastructure*.

Beneath lies a more fundamental layer:
software source code

Invisible fabric that powers our economies, administrations, and research systems.

Knowledge is in the source

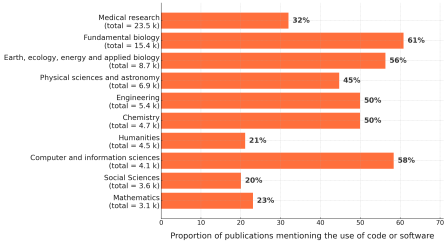
```
/**
 * @brief The basic unit of the simulation and is associated to a geographical location.
 *
 * Interventions (e.g., school closures) are tracked at this level. It contains a list of its
 * members (people), places (schools, universities, workplaces etc.), road networks, links to
 * airports etc.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

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

        int* places[MAX_NUM_PLACE_TYPES]; // list of places (of various place types) within microcell
        unsigned short int NumPlacesByType[MAX_NUM_PLACE_TYPES]; // number of places (of various place types) within mircocell
        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)

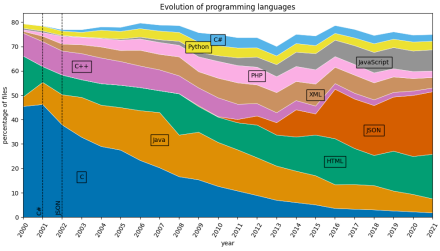
Sovranità digitale: governare dati, infrastrutture e — innanzitutto — *codice sorgente*.

A global undertaking…

Pillar of Science across all research areas



Many programming languages



From all continents

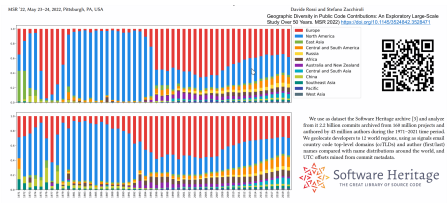
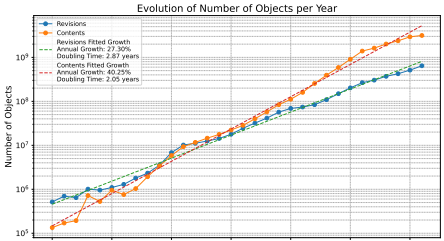


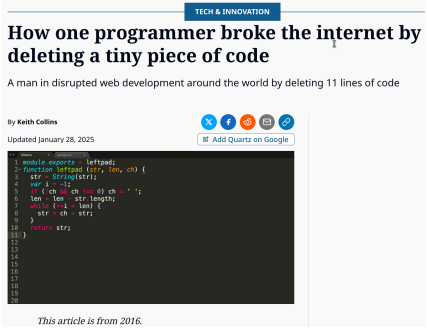
Figure 3: Ratio of commits (above) and active authors (below) by world zone over the 1971–2020 period.

An exponential growth



… and a global dependency network

An early warning: left-pad (March 2016)



TECH & INNOVATION

How one programmer broke the internet by deleting a tiny piece of code

A man in disrupted web development around the world by deleting 11 lines of code

By Keith Collins

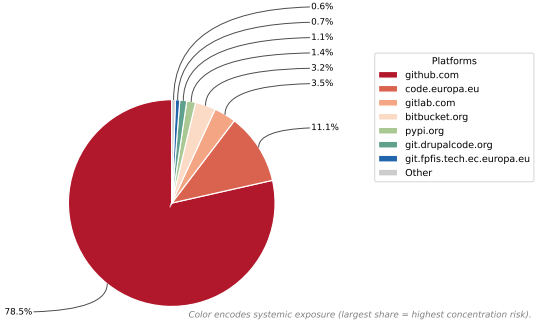
Updated January 28, 2025

Add Quartz on Google

```
module.exports = leftpad;
function leftpad (str, len, ch) {
  str = String(str);
  var i = -1;
  if (!ch && ch !== 0) ch = ' ';
  len = len - str.length;
  while (++i < len) {
    str = ch + str;
  }
  return str;
}
```

This article is from 2016.

A clear and present danger



Color encodes systemic exposure (largest share = highest concentration risk).

See the Wikipedia article:

- Facebook, PayPal, Netflix, Spotify affected
- web broken worldwide ~2.5 hours

Used by `europa.eu` devs *(Source: Software Heritage)*

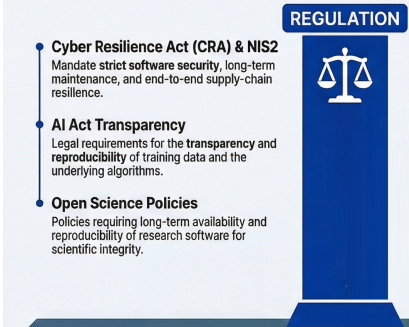
"*If GitHub or PyPI disappeared tomorrow,*
Europe would not just slow down
— *it would stop.*"

Outline

1. Why software source code matters
2. The regulation / infrastructure gap
3. Ecumenical digital sovereignty
4. Software Heritage: the ecumenical infrastructure
5. From archive to strategic backbone
6. A Very Large Telescope to explore the software development galaxy
7. We (happen to) have the answer at arm's reach
8. A path forward, together

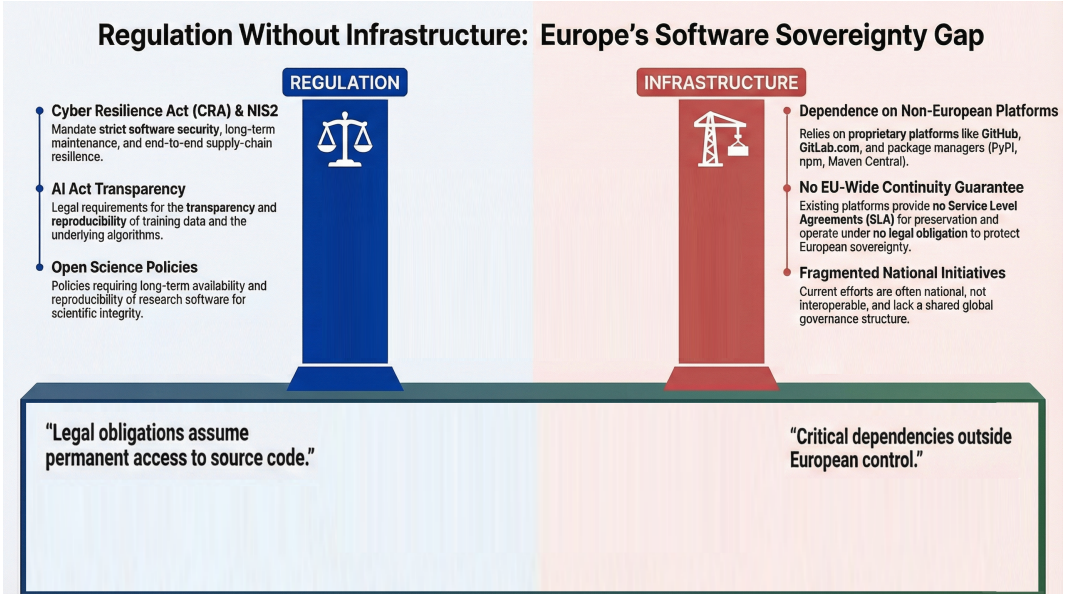
Regulation

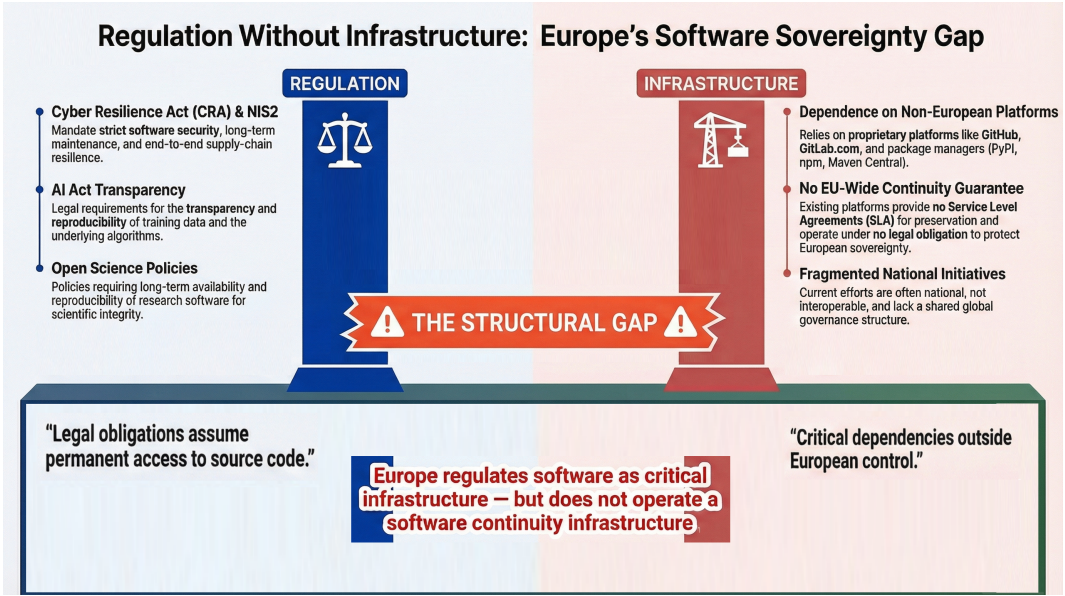
Europe's Software Sovereignty



- **Cyber Resilience Act (CRA) & NIS2**
  Mandate **strict software security**, long-term maintenance, and end-to-end supply-chain resilience.
- **AI Act Transparency**
  Legal requirements for the **transparency** and **reproducibility** of training data and the underlying algorithms.
- **Open Science Policies**
  Policies requiring long-term availability and reproducibility of research software for scientific integrity.

"Legal obligations assume permanent access to source code."


Regulation Without Infrastructure: Europe's Software Sovereignty Gap
REGULATION
Cyber Resilience Act (CRA) & NIS2
Mandate strict software security, long-term maintenance, and end-to-end supply-chain resilience.
AI Act Transparency
Legal requirements for the transparency and reproducibility of training data and the underlying algorithms.
Open Science Policies
Policies requiring long-term availability and reproducibility of research software for scientific integrity.
INFRASTRUCTURE
Dependence on Non-European Platforms
Relies on proprietary platforms like GitHub, GitLab.com, and package managers (PyPI, npm, Maven Central).
No EU-Wide Continuity Guarantee
Existing platforms provide no Service Level Agreements (SLA) for preservation and operate under no legal obligation to protect European sovereignty.
Fragmented National Initiatives
Current efforts are often national, not interoperable, and lack a shared global governance structure.
"Legal obligations assume permanent access to source code."
"Critical dependencies outside European control."


Regulation Without Infrastructure: Europe's Software Sovereignty Gap
REGULATION
Cyber Resilience Act (CRA) & NIS2
Mandate strict software security, long-term maintenance, and end-to-end supply-chain resilience.
AI Act Transparency
Legal requirements for the transparency and reproducibility of training data and the underlying algorithms.
Open Science Policies
Policies requiring long-term availability and reproducibility of research software for scientific integrity.
INFRASTRUCTURE
Dependence on Non-European Platforms
Relies on proprietary platforms like GitHub, GitLab.com, and package managers (PyPI, npm, Maven Central).
No EU-Wide Continuity Guarantee
Existing platforms provide no Service Level Agreements (SLA) for preservation and operate under no legal obligation to protect European sovereignty.
Fragmented National Initiatives
Current efforts are often national, not interoperable, and lack a shared global governance structure.
THE STRUCTURAL GAP
"Legal obligations assume permanent access to source code."
"Critical dependencies outside European control."
Europe regulates software as critical infrastructure — but does not operate a software continuity infrastructure

Outline

1. Why software source code matters
2. The regulation / infrastructure gap
3. Ecumenical digital sovereignty
4. Software Heritage: the ecumenical infrastructure
5. From archive to strategic backbone
6. A Very Large Telescope to explore the software development galaxy
7. We (happen to) have the answer at arm's reach
8. A path forward, together

Sovereignty is not isolation

What it is *not*

- it is not *autarky*
- it is not *national* silos
- it is not *forking the world*
- it is not *owning everything*

What it is

- shared, open, non-rival infrastructure
- that benefits *all* while preserving autonomy for *each*
- resilient against disruption upstream
- *ecumenical* by design

"Sovereignty is not about owning everything
— it is about ensuring that nothing essential can disappear."

Outline

1. Why software source code matters
2. The regulation / infrastructure gap
3. Ecumenical digital sovereignty
4. Software Heritage: the ecumenical infrastructure
5. From archive to strategic backbone
6. A Very Large Telescope to explore the software development galaxy
7. We (happen to) have the answer at arm's reach
8. A path forward, together

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Inria with unesco

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

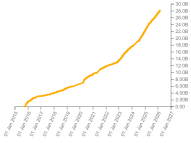
Inria with unesco

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

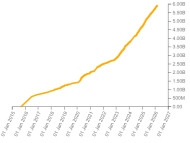
Unique digital common good *built in France since 2015*

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

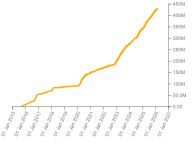
Source files
28,152,651,759



Commits
5,920,787,012



Projects
429,963,770



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

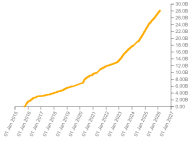
Inria with unesco

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

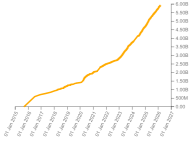
Unique digital common good *built in France since 2015*

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

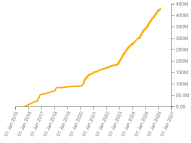
Source files
28,152,651,759

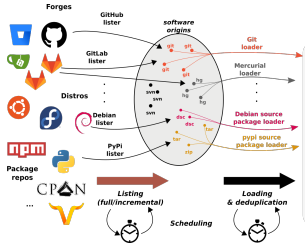


Commits
5,920,787,012



Projects
429,963,770





5000+ platforms

All versions, all history development in a single graph

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

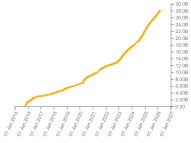
Inria with unesco

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

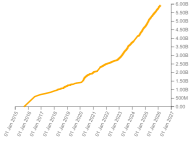
Unique digital common good *built in France since 2015*

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

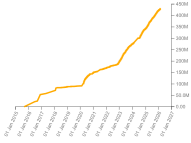
Source files
28,152,651,759

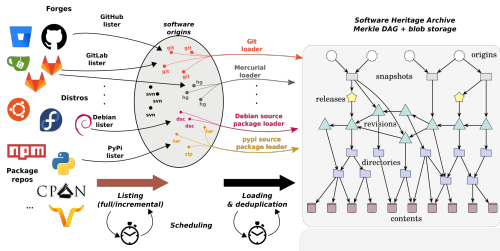


Commits
5,920,787,012



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

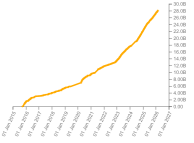
Inria with unesco

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

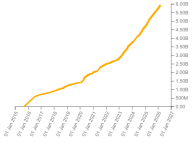
Unique digital common good *built in France since 2015*

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

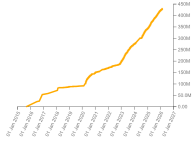
Source files
28,152,651,759

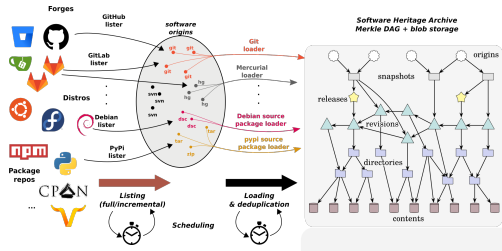


Commits
5,920,787,012



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

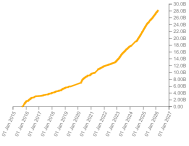
Inria with unesco

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

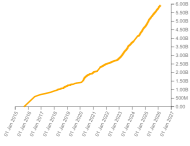
Unique digital common good *built in France since 2015*

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

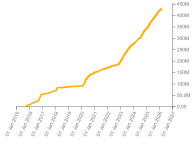
Source files
28,152,651,759

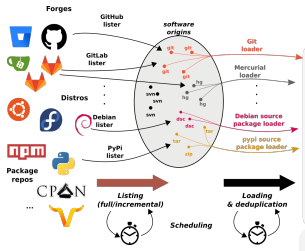


Commits
5,920,787,012



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 Canada Research Knowledge Network GitHub Premier Ministre Direction interministérielle du numérique Università di Pisa Sorbonne Université Scuola Normale Superiore Université de Lorraine Sant'Anna NWO université PARIS-SACLAY

Outline

1. Why software source code matters
2. The regulation / infrastructure gap
3. Ecumenical digital sovereignty
4. Software Heritage: the ecumenical infrastructure
5. From archive to strategic backbone
6. A Very Large Telescope to explore the software development galaxy
7. We (happen to) have the answer at arm's reach
8. A path forward, together

Selected hilights

Standards (Software Hash ID, SWHID)

Intrinsic identifier
integrity, traceability
50B+ SWHID in archive

Software Pillar of (Open) Science

EOSC SIRS report (2020)
National Open Science Plan (2021)
French National RI Roadmap (2022)

Pillar of transparent AI

StarCoder2/The Stack v2 (Hugging Face)
CodeCommons (BPI, France)
AI Factory (AI2F, EU)

Next gen Cybersecurity

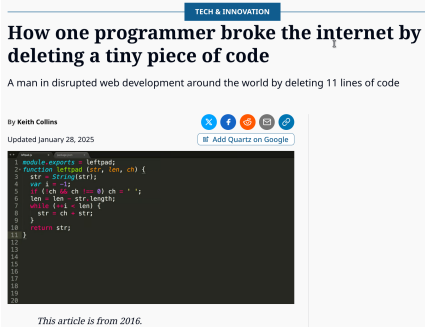
Global Vulnerabilities (Sec4AI4Sec,SWHSec)
Software supply chain (CRA, SBOM,…)

A (version of a) dependency of a critical software vanishes from GitHub,Gitlab,…:

- how to audit it five years later?
- how to reproduce the results?
- how to ensure continuity of service?
- how to retrieve them?

Back to our core digital sovereignty challenge, can we act?

An early warning: left-pad (March 2016)



TECH & INNOVATION

How one programmer broke the internet by deleting a tiny piece of code

A man in disrupted web development around the world by deleting 11 lines of code

By Keith Collins

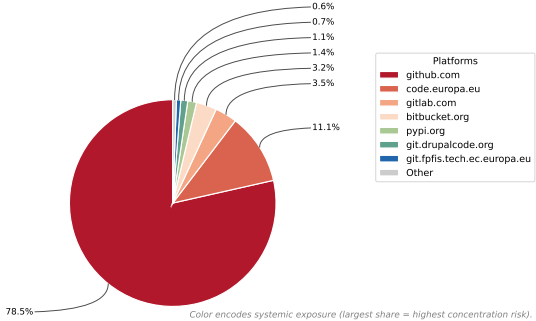
Updated January 28, 2025

Add Quartz on Google

```
module.exports = leftpad;
function leftpad (str, len, ch) {
  str = String(str);
  var i = -1;
  if (!ch && ch !== 0) ch = ' ';
  len = len - str.length;
  while (++i < len) {
    str = ch + str;
  }
  return str;
}
```

This article is from 2016.

A clear and present danger



Color encodes systemic exposure (largest share = highest concentration risk).

See the Wikipedia article:

- Facebook, PayPal, Netflix, Spotify affected
- web broken worldwide ~2.5 hours

Used by `europa.eu` devs *(Source: Software Heritage)*

"*If GitHub or PyPI disappeared tomorrow,*
Europe would not just slow down
— *it would stop.*"

Resilience, demonstrated: Guix integration

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

Connecting reproducible deployment to a long-term source code archive

Ludovic Courtès — March 29, 2019

GNU Guix can be used as a "package manager" to install and upgrade software packages as is familiar to GNU/Linux users, or as an environment manager, but it can also provision containers or virtual machines, and manage the operating system running on your machine.

One foundation that sets it apart from other tools in these areas is *reproducibility*. From a high-level view, Guix allows users to *declare* complete software environments and instantiate them. They can share those environments with others, who can replicate them or adapt them to their needs. This aspect is key to reproducible computational experiments: scientists need to reproduce software environments before they can reproduce experimental results, and this is one of the things we are focusing on in the context of the Guix-HPC effort. At a lower level, the project, along with others in the Reproducible Builds community, is working to ensure that software build outputs are reproducible, bit for bit.

Work on reproducibility at all levels has been making great progress. Guix, for instance, allows you to travel back in time. That Guix can travel back in time *and* build software reproducibly is a great step forward. But there's still an important piece that's missing to make this viable: a stable source code archive. This is where Software Heritage (SWH for short) comes in.

When source code vanishes

Guix demonstrates automated recovery using the Software Heritage archive

When a platform or registry fails… Guix fallbacks to Software Heritage and continues to build!

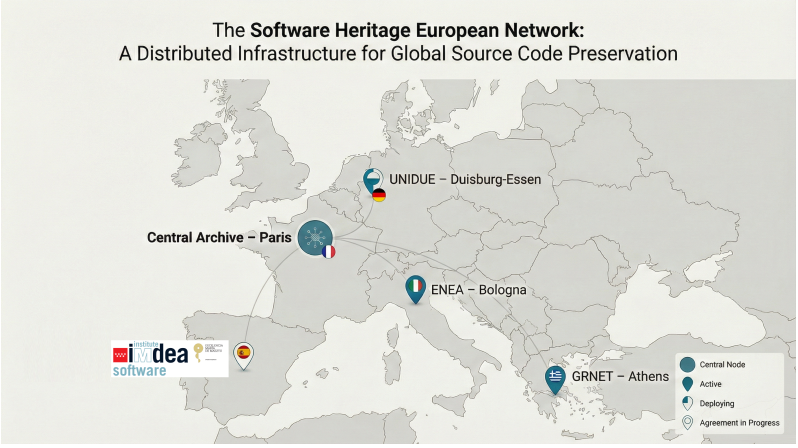
Since 2019

A sovereign fallback for Europe's build chains is possible

Extend to key package managers

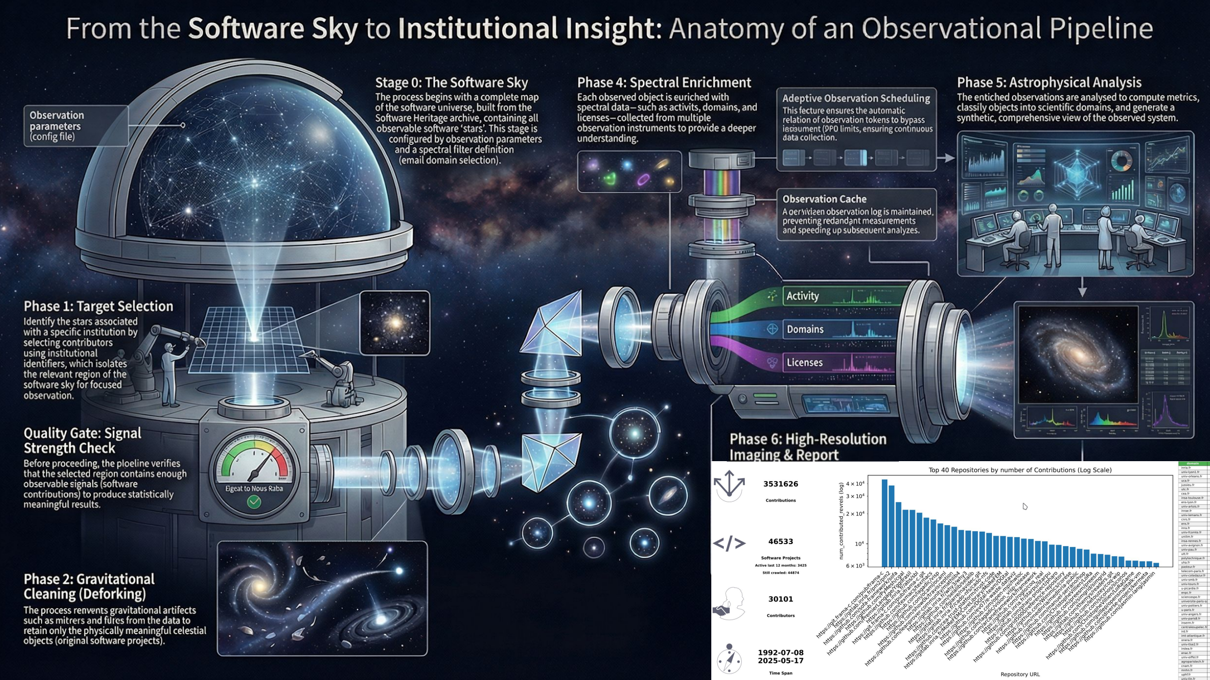
npm, PyPI, Maven Central, Cargo, CPAN, RubyGems, Packagist, …

Realize ecumenical resilience



Outline

1. Why software source code matters
2. The regulation / infrastructure gap
3. Ecumenical digital sovereignty
4. Software Heritage: the ecumenical infrastructure
5. From archive to strategic backbone
6. A Very Large Telescope to explore the software development galaxy
7. We (happen to) have the answer at arm's reach
8. A path forward, together


From the Software Sky to Institutional Insight: Anatomy of an Observational Pipeline
Observation parameters (config file)
Stage 0: The Software Sky
The process begins with a complete map of the software universe, built from the Software Heritage archive, containing all observable software 'stars'. This stage is configured by observation parameters and a spectral filter definition (email domain selection).
Phase 4: Spectral Enrichment
Each observed object is eurished with spectral data—such as activits, domains, and licenses—collected from multiple observation instruments to provide a deeper understanding.
Adeptive Observation Scheduling
This fecture ensures the automatic relation of observation tokens to bypass limits, ensuring continuous data collection.
Observation Cache
A observation log is maintained, preventing redandant measurements and speeding up subseauent analyzes.
Phase 5: Astrophysical Analysis
The entiched observations are analysed to compute metrics, classily objects into scientific domains, and generate a synthetic, comprehensive view of the observed system.
Phase 1: Target Selection
Identify the stars associated with a specific institution by selecting contributors using institutional identifiers, which isolates the relevant region of the software sky for focused observation.
Activity
Domains
Licenses
Quality Gate: Signal Strength Check
Before proceeding, the ploeline verifies that the selected region contains enough observable signals (software contributions) to produce statistically meaningful results.
Phase 6: High-Resolution Imaging & Report
3531626
Contributions
46533
Software Projects
Active last 12 months: 3425
Still crawled: 44874
30101
Contributors
1992-07-08
2025-05-17
Time Span
Top 40 Repositories by number of Contributions (Log Scale)
num_contributed_revrels (log)
Repository URL
domain
Phase 2: Gravitational Cleaning (Deforking)
The process renvents gravitational artifects such as mitrers and fures from the data to retain only the physically meaningful celestial objects (original software projects).

Technology preview: the Strategic Software Insights report

Software Heritage

Report on the public software collected, preserved and referenced for Csi-Piemonte

Version 1.0 – 2026-05-05

Contents

1 Introduction 3
- 1.1 Bird's-Eye View: the key numbers 3
- 1.2 Technical Information 3

2 Analysis of Contributions 5
- 2.1 Density of contributions per project 5
- 2.2 Distribution of first contributions over time 7
- 2.3 Timespan of institutional contributions 9
- 2.4 Distribution of project lifetime 11
- 2.5 Distribution of Citation Files 13
  - 2.5.1 Projects with Citation Files 13
- 2.6 Top Projects by Repository Host 14

3 Analysis of the Projects to which Csi-Piemonte contributes 31
- 3.1 Distribution by number of stars 31
- 3.2 Main organizations 33
- 3.3 Top 40 projects by number of contributions 34
- 3.4 Top 40 projects by popularity 35
- 3.5 Top 40 projects by repository age 37

4 Programming Languages 54

5 License Analysis 56
- 5.1 License Distribution 56
- 5.2 License Status Overview 56
- 5.3 License Statistics 56
- 5.4 Top License Types 56

6 Platform Distribution Analysis 58
- 6.1 Platform Distribution Overview 58
- 6.2 Platform Distribution (Log Scale) 58
- 6.3 Platform Statistics 58
- 6.4 Top Platforms 58
- 6.5 Summary 58

7 Tentative Classification in Research Areas/Domains 60
- 7.1 Distribution of domains of contributors (all projects) 61
- 7.2 Distribution of domains of contributors (selected projects with more than 1 contributions) 61

8 Methodology: From Source Graph to Analysis-Ready Dataset 61

9 Conclusion 62

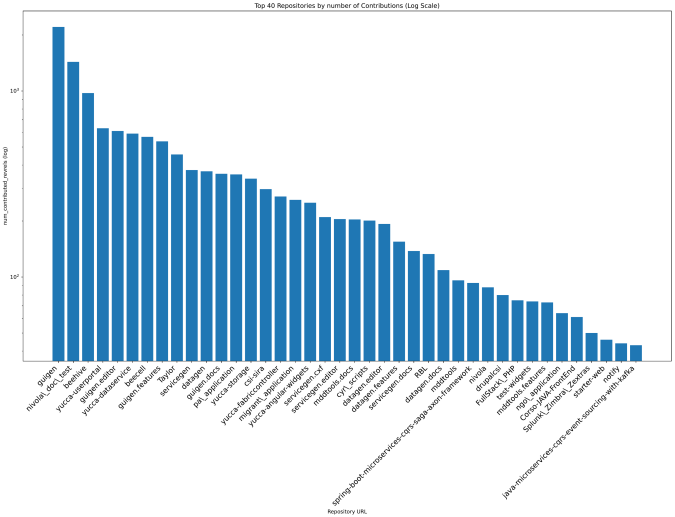
A peek inside the CSI Piemonter Strategic Insights

15472
Contributions

392
Software Projects
Active last 12 months: 17
Still crawled: 365

115
Contributors

2000-01-01
2026-02-26
Time Span



Domain	Contributors

Outline

1. Why software source code matters
2. The regulation / infrastructure gap
3. Ecumenical digital sovereignty
4. Software Heritage: the ecumenical infrastructure
5. From archive to strategic backbone
6. A Very Large Telescope to explore the software development galaxy
7. We (happen to) have the answer at arm's reach
8. A path forward, together

Software Heritage: one infrastructure, six strategic levers

Research infrastructure

two facets:

- support Open Science
- enable Software Science

Traceability and compliance

- Software supply chain (SBOM)
- integrity (SWHID)
- availability
- support CRA/NIS2 mandates

Cybersecurity

Cyber forensics, vulnerability tracking at world scale

Resilience

when npm/PyPI/Maven fail,
Europe continues to build

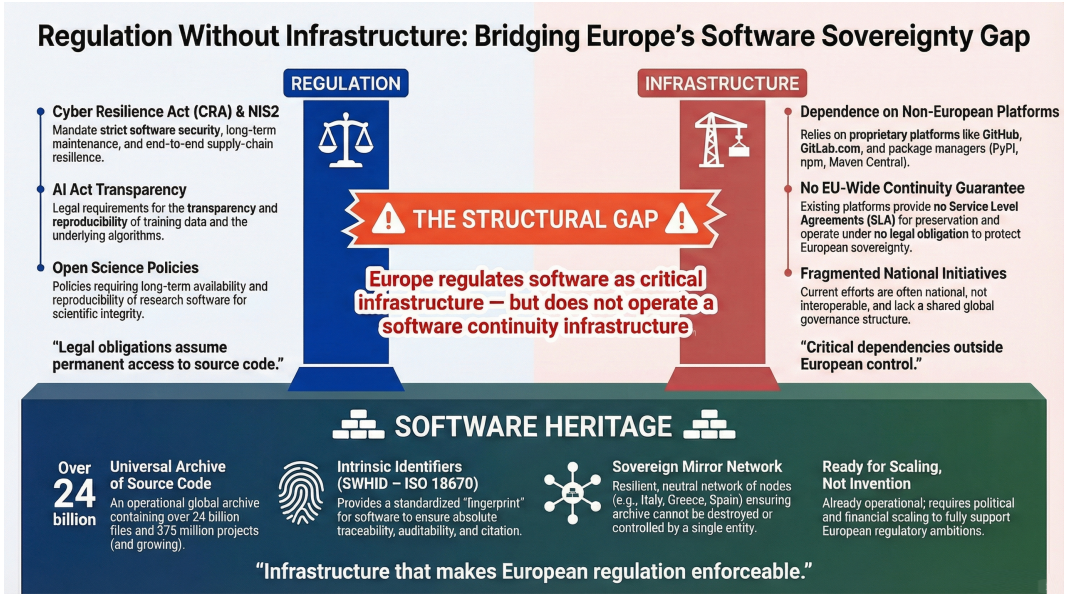
Transparent AI

- CodeCommons : provenance and attributon for LLMs
- AI Act, Art. 11, 12, 15:
  - technical documentation
  - record-keeping
  - reproducibility

Metrics we can count on

- Objective Strategic Insights
- Support policy implementation

The missing foundation to fill the gap is here



Outline

1. Why software source code matters
2. The regulation / infrastructure gap
3. Ecumenical digital sovereignty
4. Software Heritage: the ecumenical infrastructure
5. From archive to strategic backbone
6. A Very Large Telescope to explore the software development galaxy
7. We (happen to) have the answer at arm's reach
8. A path forward, together

What this means for Italy and for Europe

Europe does not need to invent this infrastructure…

it can adopt and scale what took 10+ years to build.

Concrete next steps

All recognise software continuity as a critical enabler for

- digital sovereignty
- regulation enforcement (CRA, NIS2, AI Act)
- research, education and open science

All avoid balkanisation, support mutualisation

- build on common, shared, open, non-profit infrastructures

Public administrations and enterprises *join* as members, mirror operators, contributors

- turn *dependence* into *shared resilience*
- position Italy as a leader in this ecumenical approach

True digital sovereignty is not control — it is resilience.

A natural foundation for the Decalogo per la Sovranità Digitale.

https://www.softwareheritage.org