

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Opening remarks, and lessons from Software Heritage

Roberto Di Cosmo · Paris Peace Forum, Inria Paris, 9 November 2026

Hosted by





In collaboration with



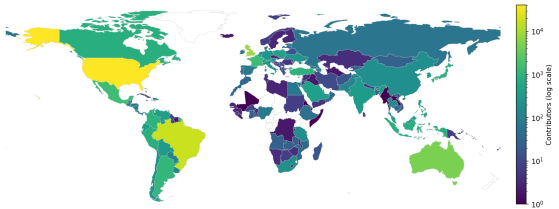
State of Public Code 2026: what governments actually contribute

Public Code Observatory: United Nations (ODET), UNESCO, Software Heritage. Edition 2026-10, 28 October 2026. Preliminary, open to comments from all Member States.

- **240 922** projects with government contributions, **82 438** people
- **193** Member States and 2 observers; **164** with measured contributions
- **90.8 %** sits on a single platform
- best predictor is not wealth but **digital-government development**
- about **4 in 5** projects stop within a year

Whole archive, not a sample: every figure recomputable by anyone.

stateofpubliccode.org

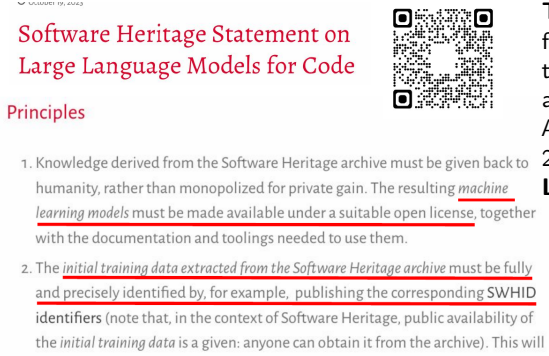


government contributors per country (log scale)



cross-border collaboration: shared projects per country pair

Software Heritage meets generative AI: principles, then lessons



October 19, 2023

Software Heritage Statement on Large Language Models for Code

Principles

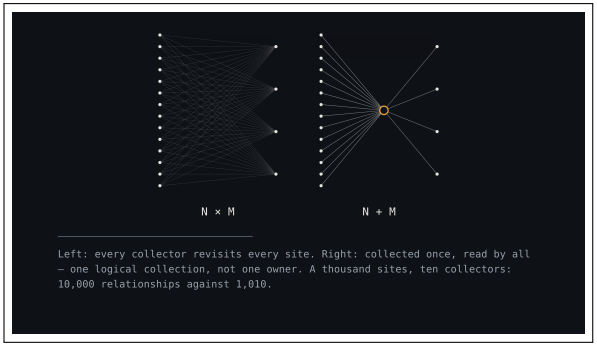
1. Knowledge derived from the Software Heritage archive must be given back to humanity, rather than monopolized for private gain. The resulting *machine learning models* must be made available under a suitable open license, together with the documentation and toolings needed to use them.
2. The *initial training data extracted from the Software Heritage archive* must be fully and precisely identified by, for example, publishing the corresponding SWHID identifiers (note that, in the context of Software Heritage, public availability of the *initial training data* is a given: anyone can obtain it from the archive). This will enable use cases such as: studying biases (fairness), verifying if a code of interest was present in the training data (transparency), and providing appropriate attribution when generated code bears resemblance to training data (credit), among others.
3. Mechanisms should be established, where possible, for authors to exclude their archived code from the training inputs before model training begins.

Three principles, October 2023: knowledge derived from the archive is **given back** as open models; training data is **fully identified** by its **SWHIDs**; authors can **exclude** their code before training. Applied in **The Stack v2** and **StarCoder2**, February 2024.

Lessons learned

- **Transparency is easy**: an identifier (SWHID, ISO/IEC 18670:2025) and the archive, no petabytes to ship.
- **Reuse rights are tricky**: who owns a source file, under which terms?
- **A qualified training set is expensive**: licence detection at massive scale.
- **Attribution on model output is hard.**
- Hence **CodeCommons**, a coordinated effort, and **AIPREF** at the IETF: preferences that survive copying a file.

Good enough is not good enough: how data is collected today



every collector revisits every site, against one logical collection read by all

Series and sources, every claim with its reference:
dicosmo.org/good-enough

Four challenges

- **Collection is N × M.** Every crawler revisits every site, and the open web closes an API and a bot filter at a time.
- **Good enough wins because the cost is an externality.** Each actor is reasonable, the aggregate is damaging, nobody owns the aggregate.
- **The walls stop the legible traffic.** Filters block the crawler that identifies itself, and the archive's own users.
- **The alternative was written down in 1998.** One logical collection, many copies, one identifier recomputable by anyone.

These slides:
`annex.softwareheritage.org/public/talks/2026/2026-11-09-ppf-inria-roundtable2.pdf`