Outline

1. Software and Open Science
2. An emerging software pillar of Open Science
3. (Open Source) Software Supply Chain
4. Addressing shared needs: meet Software Heritage
5. Conclusion
Software is a pillar of Open Science

Software powers modern research

[...] software [...] essential in their fields.
Top 100 papers (Nature, 2014)

Sometimes, if you don’t have the software, you don’t have the data
Christine Borgman, Paris, 2018

A key pillar: software (source code)

The links in the picture are important

Nota Bene

software may be a tool, a research outcome and a research object
access to the source code is essential!

Preserving (the history of) source code is necessary for reproducibility
Software Source Code is Precious Knowledge

“Programs must be written for people to read, and only incidentally for machines to execute.”

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  BIT6  # IS THE LR ANTENNA IN POSITION 1 YET
         EXTEND
         RAND  CHAN33
         EXTEND
         BZF  P63SP0T4  # BRANCH IF ANTENNA ALREADY IN POSITION 1
         CAF  CODE5000  # ASTRONAUT: PLEASE CRANK THE
         TC   BANKCALL  # SILLY THING AROUND
         CADR  GOPERF1
         TCF  GOTOPOOH  # TERMINATE
         TCF  P63SP0T3  # PROCEED  SEE IF HE'S LYING

P63SP0T4  TC  BANKCALL  # ENTER  INITIALIZE LANDING RADAR
         CADR  SETPOS1
         TC  POSTJUMP  # OFF TO SEE THE WIZARD ...
```

Quake III source code (excerpt)

```
float Q_rsqr( 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.”

R. Di Cosmo roberto@dicosmo.org (CC-BY 4.0)
How are we managing our software?

Reproducibility, maintenance in Academia

Security, integrity, traceability in Industry

Can they track the software that they ship, use, acquire has that bug or vulnerability

Awareness is raising at the level of public policy

(articles: here, here, here and here)
French National plan for Open Science, 2021-2024

Path Three: Opening up and promoting source code produced by research

7. Recognize and support the dissemination under an open source license of software produced by publicly funded research programmes

8. Highlight the production of source code from higher education, research and innovation

9. Define and promote an open source software policy

« The opening of software source code is a major challenge for the reproductibility of scientific results. »

« Distribution of software products under open source licence will be preferred. »

- 129 projects
- 4 awards
- 6 accessit
- first edition

Remise des prix science ouverte du logiciel libre de la recherche

Le ministère de l’Enseignement supérieur, de la Recherche et de l’Innovation remet pour la première année les Prix science ouverte du logiciel libre de la recherche. Dix logiciels mis au point par des équipes françaises sont récompensés pour leur contribution à l’avancement de la connaissance scientifique.
<table>
<thead>
<tr>
<th>Archive</th>
<th>Research software artifacts must be properly <strong>archived</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>make sure we can <strong>retrieve</strong> them (<strong>reproducibility</strong>).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
<th>Research software artifacts must be properly <strong>referenced</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>make sure we can <strong>identify</strong> them (<strong>reproducibility</strong>).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Describe</th>
<th>Research software artifacts must be properly <strong>described</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>make it easy to <strong>discover</strong> and <strong>reuse</strong> them (<strong>visibility</strong>).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cite/Credit</th>
<th>Research software artifacts must be properly <strong>cited</strong> (not the same as referenced!)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>to give <strong>credit</strong> to authors (<strong>evaluation</strong>).</td>
</tr>
</tbody>
</table>

These are also **industry** needs!
Open Source is growing...

Software is eating the world

**The Wall Street Journal**

**Essay**

*Why Software Is Eating The World*

By Marc Andreessen

August 20, 2011

This week, Hewlett-Packard (where I am on the board) announced that it is exploring jettisoning its struggling PC business in favor of investing more heavily in software, where it sees better potential for growth. Meanwhile, Google plans to buy up the cellphone handset maker Motorola Mobility. Both moves surprised the tech world. But both moves are also in line with a trend I’ve observed, one that makes me optimistic about the future.

Software companies outperform or buy out traditional companies

Marc Andreessen, 2011

Reuse is the new rule

80% to 90% of a new application is ... just reuse!

(Sonatype survey, 2017)
Software supply chain integrity

Can you track the software that...
- you ship
- you use
- you acquire
- has that bug
- has that vulnerability

KYSW: Know Your SoftWare - like KYC in banking

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

Can we fulfil together these shared needs?
Outline

1. Software and Open Science
2. An emerging software pillar of Open Science
3. (Open Source) Software Supply Chain
4. Addressing shared needs: meet Software Heritage
5. Conclusion
Software supply chain and its issues

Complex digital infrastructure

Software supply chain in the news

So/f_tware Supply Chain attacks

Malicious code injection into software components to compromise downstream users

March 2022  node-ipc and peacenotwar (CVE-2022-23812)

Dec 2021  Apache Log4j Remote Code Execution (Log4Shell, CVE-2021-44228)

Nov 2018  Attack on NPM package event-stream
Software supply chain in a picture
A long road ahead

Vertical approach
improve security of *each component* separately

Horizontal approach
explore *the whole supply chain*

A few key challenging properties

- **findability** needs qualified metadata
- **availability** needs an archive and a system of identifiers
- **integrity** needs crypto
- **traceability** needs a global provenance database
- **reproducibility** needs groundbreaking tools

We need a *global coordinated effort*…
and a *common, open, shared* infrastructure to track *all (Open Source) software!*
Forges are key platforms, but they are not enough!

**2015: the first big bad news**

Google Code and Gitorious.org shutdown: ~1M endangered repositories
- broken links in the web of knowledge (my papers too)

**Big bad news keep coming in**

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250.000+ repositories (including research software)
- summer 2022: GitLab.com considers erasing all projects that are inactive for a year

**In Academia too!**

- 2021: Inria’s old gforge is unplugged… breaks the Opam build chain for OCaml

source code is spread across hundreds of them…
- lack of uniformity, no persistence guarantee
Outline

1. Software and Open Science
2. An emerging software pillar of Open Science
3. (Open Source) Software Supply Chain
4. Addressing shared needs: meet Software Heritage
5. Conclusion
Software Heritage, in a nutshell

www.softwareheritage.org

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
Universal software archive, principled


One infrastructure
open and shared

Technology
- transparency and FOSS
- replicas all the way down

Content (billions!)
- intrinsic identifiers
- facts and provenance

Organization
- non-profit
- multi-stakeholder

Largest archive

R. Di Cosmo roberto@dicosmo.org (CC-BY 4.0)
An international, non profit initiative built for the long term

Sharing the vision

And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors

Diamond sponsor

Platinum sponsors

Gold sponsors

Silver sponsors

Bronze sponsors
Global development history permanently archived in a uniform data model

- over 14 billion unique source files from over 210 million software projects
- ~1PB (compressed) blobs, ~30 B nodes, ~400 B edges
**Intrinsic Identifiers for software artefacts**

Software Heritage Identifiers (SWHID) link to full docs

25+B **intrinsic, decentralised, cryptographically strong identifiers, SWHIDs**

Emerging standard: Linux Foundation SPDX 2.2; IANA registered; WikiData P6138

Full fledged *source code references* for reproducibility

Examples: Apollo 11 AGC excerpt, Quake III rsqrt; Guidelines available, see ICMS 2020
A quick tour

- Browse (e.g. Apollo 11, and your work may be already there !)
- Trigger archival, use the updateswh browser extension (GitHub action available too)
- Get and use SWHIDs (full specification available online)
- Cite software with biblatex-software package from CTAN
  - Overleaf ACMART template available
- Example in journals: article from IPOL
- Example with Parmap: devel on Github, archive in SWH, curated deposit in HAL
- Extracting all the software products for Inria, for CNRS, for CNES, for LIRMM or for Rémi Gribonval using HalTools
- Curated deposit in SWH via HAL, see for example: LinBox, SLALOM, Givaro, NS2DDV, SumGra, Coq proof, …
- Example use in research articles:
  - compare Fig. 1 and conclusions in the 2012 version and the updated version
  - SWHID in a replication experiment
A revolutionary infrastructure

The **graph** of Software Development

All software development in a single graph ...  
- enable traceability

The **blockchain** of Software Development

... 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, ...
Outline

1. Software and Open Science
2. An emerging software pillar of Open Science
3. (Open Source) Software Supply Chain
4. Addressing shared needs: meet Software Heritage
5. Conclusion
A rally flag for a grand vision

Bring together academia, industry, governments, communities

"to build a reference, global infrastructure for open and better software"

Software Heritage is the first brick …

- vendor neutral
- open source
- 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!

fund and/or develop SWH, use SWH research, build tools, contribute to swhid.org
The floor is yours

Let’s all work together!

Questions?

References