

Towards a Software Pillar for Open Science

challenges and opportunities

Roberto Di Cosmo
Inria and Université de Paris

25 October 2021
ECSS



Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

- 1 Introduction
- 2 Open Science
- 3 Building the software pillar of Open Science: assessing the needs
- 4 Phase 1: focus on ARDC and infrastructures
- 5 Demo time!
- 6 Phase 2: broader policy issues



Computer Science professor in Paris, now working at INRIA

- 30 years of research (Theor. CS, Programming, Software Engineering, Erdos #: 3)
- 20 years of Free and Open Source Software
- 10 years building and directing structures for the common good



1999 *DemoLinux* – first live GNU/Linux distro

2007 *Free Software Thematic Group*
150 members 40 projects 200Me

2008 *Mancoosi project* www.mancoosi.org

2010 *IRILL* www.irill.org

2015 *Software Heritage* at INRIA

2018 *National Committee for Open Science*, France

- 1 Introduction
- 2 Open Science
- 3 Building the software pillar of Open Science: assessing the needs
- 4 Phase 1: focus on ARDC and infrastructures
- 5 Demo time!
- 6 Phase 2: broader policy issues



Why Open Science?

Open Science ([Second National Plan for Open Science](#), France, 2021)

Unhindered dissemination of results, methods and products from scientific research. It draws on *the opportunity provided by recent digital progress* to develop open access to *publications* and – as much as possible – *data, source code and research methods*.

Why Open Science?

Open Science ([Second National Plan for Open Science](#), France, 2021)

Unhindered dissemination of results, methods and products from scientific research. It draws on *the opportunity provided by recent digital progress* to develop open access to *publications* and – as much as possible – *data, source code and research methods*.

Jean-Eric Paquet (EU DGRI, [on the objective of Open Science](#))

“Increase *scientific quality*, the *pace of discovery and technological development*, as well as *societal trust in science*.”

Why Open Science?

Open Science ([Second National Plan for Open Science](#), France, 2021)

Unhindered dissemination of results, methods and products from scientific research. It draws on *the opportunity provided by recent digital progress* to develop *open access to publications* and – as much as possible – *data, source code and research methods*.

Jean-Eric Paquet (EU DGRI, [on the objective of Open Science](#))

“Increase *scientific quality*, the *pace of discovery and technological development*, as well as *societal trust in science*.”

Mariya Gabriel ([EU Commissioner](#) for Research)

The COVID-19 crisis has also shown that cooperation at international level in research and innovation is more important than ever, including through *open access to data and results*. *No nation, no country can tackle any of these global challenges alone*.

Why Open Science?

Open Science ([Second National Plan for Open Science](#), France, 2021)

Unhindered dissemination of results, methods and products from scientific research. It draws on *the opportunity provided by recent digital progress* to develop *open access to publications* and – as much as possible – *data, source code and research methods*.

Jean-Eric Paquet (EU DGRI, [on the objective of Open Science](#))

“Increase *scientific quality*, the *pace of discovery and technological development*, as well as *societal trust in science*.”

Mariya Gabriel ([EU Commissioner](#) for Research)

The COVID-19 crisis has also shown that cooperation at international level in research and innovation is more important than ever, including through *open access to data and results*. *No nation, no country can tackle any of these global challenges alone*.

Yuval Noah Harari (on COVID 19)

“*The real antidote [to epidemic] is scientific knowledge and global cooperation*.”

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

19XX's compulsory exclusive copyright transfer to publishers (unlawful?)

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

19XX's compulsory exclusive copyright transfer to publishers (unlawful?)
(notable exceptions: [US federal agencies](#) and [UK Crown Copyright](#))

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

- 19XX's compulsory exclusive copyright transfer to publishers (unlawful?)
(notable exceptions: [US federal agencies](#) and [UK Crown Copyright](#))
- 1990's Internet, Web and ArXiv break the [marriage of convenience of researchers with publishers](#)

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

- 19XX's compulsory exclusive copyright transfer to publishers (unlawful?)
(notable exceptions: [US federal agencies](#) and [UK Crown Copyright](#))
- 1990's Internet, Web and ArXiv break the [marriage of convenience of researchers with publishers](#)
- 2000's declarations (Budapest, 2001; Berlin 7, 2009) and actions (LIPIcs, 2009)

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

- 19XX's compulsory exclusive copyright transfer to publishers (unlawful?)
(notable exceptions: [US federal agencies](#) and [UK Crown Copyright](#))
- 1990's Internet, Web and ArXiv break the [marriage of convenience of researchers with publishers](#)
- 2000's declarations (Budapest, 2001; Berlin 7, 2009) and actions (LIPIcs, 2009)
- 2010's reactions (SciHub, 2011; [Plan S](#), 2018) and transformations ([not so easy](#))

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

- 19XX's compulsory exclusive copyright transfer to publishers (unlawful?)
(notable exceptions: [US federal agencies](#) and [UK Crown Copyright](#))
- 1990's Internet, Web and ArXiv break the [marriage of convenience of researchers with publishers](#)
- 2000's declarations (Budapest, 2001; Berlin 7, 2009) and actions (LIPIcs, 2009)
- 2010's reactions (SciHub, 2011; [Plan S](#), 2018) and transformations ([not so easy](#))
TL;DR: see [my viewpoint in 2005](#) and [the SIGPLAN blog in 2020](#)

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

- 19XX's compulsory exclusive copyright transfer to publishers (unlawful?) (notable exceptions: [US federal agencies](#) and [UK Crown Copyright](#))
 - 1990's Internet, Web and ArXiv break the [marriage of convenience of researchers with publishers](#)
 - 2000's declarations (Budapest, 2001; Berlin 7, 2009) and actions (LIPIcs, 2009)
 - 2010's reactions (SciHub, 2011; [Plan S](#), 2018) and transformations ([not so easy](#))
- TL;DR: see [my viewpoint in 2005](#) and [the SIGPLAN blog in 2020](#)

Open Data (much less painful story)

- 1957-1958: International Geophysical Year shows the way

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

- 19XX's compulsory exclusive copyright transfer to publishers (unlawful?) (notable exceptions: [US federal agencies](#) and [UK Crown Copyright](#))
 - 1990's Internet, Web and ArXiv break the [marriage of convenience of researchers with publishers](#)
 - 2000's declarations (Budapest, 2001; Berlin 7, 2009) and actions (LIPIcs, 2009)
 - 2010's reactions (SciHub, 2011; [Plan S](#), 2018) and transformations ([not so easy](#))
- TL;DR: see [my viewpoint in 2005](#) and [the SIGPLAN blog in 2020](#)

Open Data (much less painful story)

- 1957-1958: International Geophysical Year shows the way
- 2006 (and 2021): OECD recommendation on [publicly funded research data](#)

Two well known pillars of Open Science

Open Access (a long, painful, unfinished story)

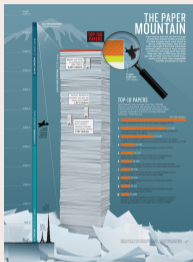
- 19XX's compulsory exclusive copyright transfer to publishers (unlawful?) (notable exceptions: [US federal agencies](#) and [UK Crown Copyright](#))
 - 1990's Internet, Web and ArXiv break the [marriage of convenience of researchers with publishers](#)
 - 2000's declarations (Budapest, 2001; Berlin 7, 2009) and actions (LIPIcs, 2009)
 - 2010's reactions (SciHub, 2011; [Plan S](#), 2018) and transformations ([not so easy](#))
- TL;DR: see [my viewpoint in 2005](#) and [the SIGPLAN blog in 2020](#)

Open Data (much less painful story)

- 1957-1958: International Geophysical Year shows the way
- 2006 (and 2021): OECD recommendation on [publicly funded research data](#)
- 2016 and later: FAIR terminology (*focus on metadata, sort of forgets open...*)

A long overlooked pillar of Open Science

Software powers modern research



[...] software [...] essential in their fields.

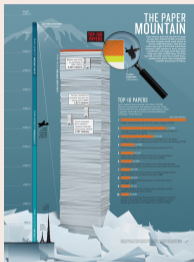
Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data

Christine Borgman, Paris, 2018

A long overlooked pillar of Open Science

Software powers modern research



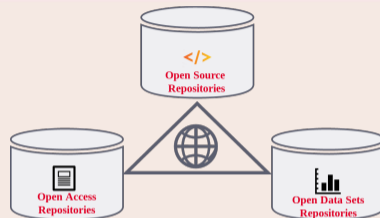
[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data

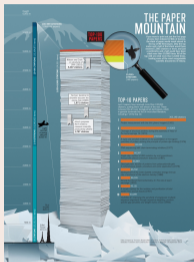
Christine Borgman, Paris, 2018

Missing pillar: software (source code)



A long overlooked pillar of Open Science

Software powers modern research



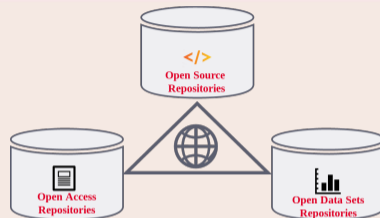
[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data

Christine Borgman, Paris, 2018

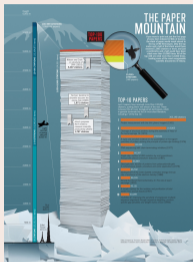
Missing pillar: software (source code)



The links in the picture are **important**

A long overlooked pillar of Open Science

Software powers modern research



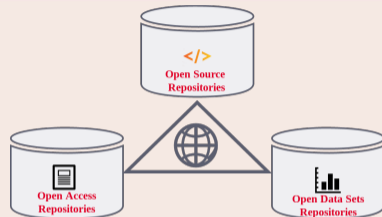
[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data

Christine Borgman, Paris, 2018

Missing pillar: software (source code)



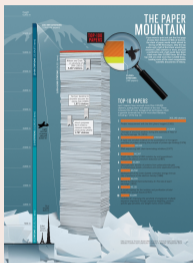
The links in the picture are **important**

Nota Bene

software may be a *tool*, a *research outcome* and a *research object*

A long overlooked pillar of Open Science

Software powers modern research



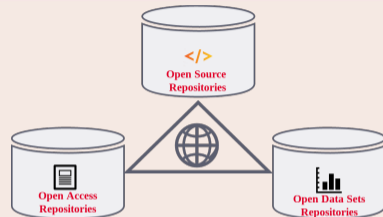
[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data

Christine Borgman, Paris, 2018

Missing pillar: software (source code)



The links in the picture are **important**

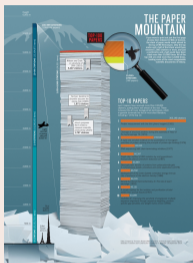
Nota Bene

software may be a *tool*, a *research outcome* and a *research object*

access to the *source code* is essential!

A long overlooked pillar of Open Science

Software powers modern research



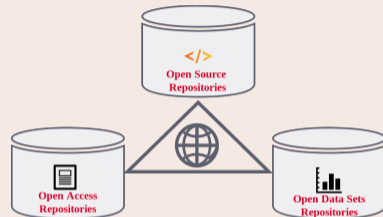
[...] software [...] essential in their fields.

Top 100 papers (Nature, 2014)

Sometimes, if you don't have the software, you don't have the data

Christine Borgman, Paris, 2018

Missing 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

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 *Source Code* is Precious Knowledge

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      CODE500      # ASTRONAUT:  PLEASE CRANK THE
              TC       BANKCALL      #              SILLY THING AROUND
              CADR     GOPERF1
              TCF      GOTOP00H      # 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 ...
              CADR     BURNBABY
```

Software Source Code is Precious Knowledge

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    CODE500      # ASTRONAUT: PLEASE CRANK THE
              TC     BANKCALL     # SILLY THING AROUND
              CADR   GOPERF1
              TCF    GOTOP00H     # 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 ...
              CADR   BURNBABY
```

Quake III source code (excerpt)

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

    return y;
}
```

Software Source Code is Precious Knowledge

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       CODE500      # ASTRONAUT: PLEASE CRANK THE
              TC        BANKCALL     # SILLY THING AROUND
              CADR      GOPERF1
              TCF       GOTOP00H     # 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 ...
              CADR      BURNBABY
```

Quake III source code (excerpt)

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 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 * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalfs - ( 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.”

Source code is *special* (software is *not* data)

Software *evolves* over time

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

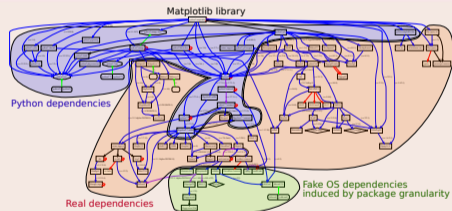
Source code is *special* (software is *not* data)

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*



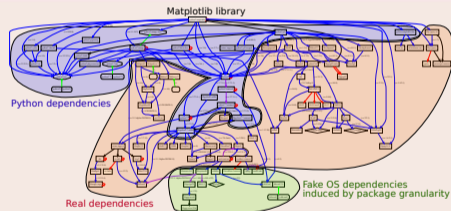
Source code is *special* (software is *not* data)

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*



The human side

design, algorithm, code, test, documentation, community, funding

and so many more facets ...

The Paris Call on Software Source code (2019, UNESCO)

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 to meet in Paris

The Paris Call on Software Source code (2019, UNESCO)

Experts call for greater recognition of software source code as heritage for sustainable development

16 November 2018



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



The call is published on Feb 2019

The Paris Call on Software Source code (2019, UNESCO)

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 to meet in Paris



The call is published on Feb 2019

“[We call to] promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, sharing good practices and recognising in the careers of academics their contributions to high quality software development, in all their forms”

<https://en.unesco.org/foss/paris-call-software-source-code>

- 1 Introduction
- 2 Open Science
- 3 Building the software pillar of Open Science: assessing the needs**
- 4 Phase 1: focus on ARDC and infrastructures
- 5 Demo time!
- 6 Phase 2: broader policy issues



Researchers

- **archive** and **reference** software used in articles
- **find** useful software
- get **credit** for developed software
- verify, **reproduce**, improve results

A plurality of needs

Researchers

- **archive** and **reference** software used in articles
- **find** useful software
- get **credit** for developed software
- verify, **reproduce**, improve results

Laboratories/teams

- **track** software contributions
- produce reports
- maintain web page

A plurality of needs

Researchers

- **archive** and **reference** software used in articles
- **find** useful software
- get **credit** for developed software
- verify, **reproduce**, improve results

Laboratories/teams

- **track** software contributions
- produce reports
- maintain web page

Research Organization

know its **software assets**

- technology **transfer**
- impact **metrics**
- funding **strategy**
- career **evaluation**

Archive

Research software artifacts must be properly **archived**
make sure we can *retrieve* them (*reproducibility*)

Archive

Research software artifacts must be properly **archived**
make sure we can *retrieve* them (*reproducibility*)

Reference

Research software artifacts must be properly **referenced**
make sure we can *identify* them (*reproducibility*)

Archive

Research software artifacts must be properly **archived**
make sure we can *retrieve* them (*reproducibility*)

Reference

Research software artifacts must be properly **referenced**
make sure we can *identify* them (*reproducibility*)

Describe

Research software artifacts must be properly **described**
make it easy to *discover* and *reuse* them (*visibility*)

Archive

Research software artifacts must be properly **archived**
make sure we can *retrieve* them (*reproducibility*)

Reference

Research software artifacts must be properly **referenced**
make sure we can *identify* them (*reproducibility*)

Describe

Research software artifacts must be properly **described**
make it easy to *discover* and *reuse* them (*visibility*)

Cite/Credit

Research software artifacts must be properly **cited** (*not the same as referenced!*)
to give *credit* to authors (*evaluation!*)

What is at stake: beyond ARDC

Policy framework for dissemination, reuse, evaluation and recognition

Define and promote an open source policy for publicly funded research software, including incentives and recognition for researchers and engineers

What is at stake: beyond ARDC

Policy framework for dissemination, reuse, evaluation and recognition

Define and promote an open source policy for publicly funded research software, including incentives and recognition for researchers and engineers

Sustainability

Organisational schemas, legal tools, economic models, processes and policies to ensure research software can be maintained and sustained over time

What is at stake: beyond ARDC

Policy framework for dissemination, reuse, evaluation and recognition

Define and promote an open source policy for publicly funded research software, including incentives and recognition for researchers and engineers

Sustainability

Organisational schemas, legal tools, economic models, processes and policies to ensure research software can be maintained and sustained over time

Technology transfer and industry collaboration

Approaches, support, methods, processes to establish connections with industry in order to foster uptake and transfer of research software

What is at stake: beyond ARDC

Policy framework for dissemination, reuse, evaluation and recognition

Define and promote an open source policy for publicly funded research software, including incentives and recognition for researchers and engineers

Sustainability

Organisational schemas, legal tools, economic models, processes and policies to ensure research software can be maintained and sustained over time

Technology transfer and industry collaboration

Approaches, support, methods, processes to establish connections with industry in order to foster uptake and transfer of research software

Advanced technologies and tools

software quality reproducibility, and traceability (including plagiarism detection)

- 1 Introduction
- 2 Open Science
- 3 Building the software pillar of Open Science: assessing the needs
- 4 Phase 1: focus on ARDC and infrastructures**
- 5 Demo time!
- 6 Phase 2: broader policy issues



The state of the art (in CS!) is far from ideal

ICSE (Zannier, Melrik, Maurer, 2006)

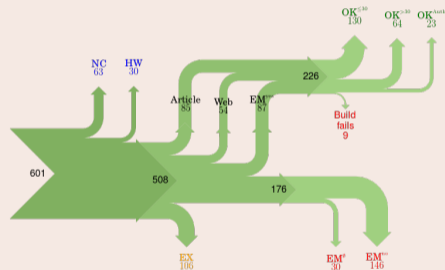
absence of replication studies

ACM TOSEM 2001 to 2006

C. Ghezzi

60% papers with tools: **only 20% installable**

Collberg's 2015 reproducibility study



601 mainstream papers

- 508 with tools
- **only 40% installable**

The state of the art (in CS!) is far from ideal

ICSE (Zannier, Melrik, Maurer, 2006)

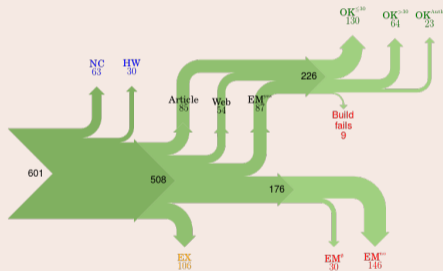
absence of replication studies

ACM TOSEM 2001 to 2006

C. Ghezzi

60% papers with tools: **only 20% installable**

Collberg's 2015 reproducibility study



Main reasons: source code (*or the right version of it*) cannot be found

- **policy issue**: opening up the code of research software
- **infrastructures**: archive and reference it

let's start here

Where is the source code?

Collaborative development platforms (aka "forges")

- BitBucket, GitLab(.com), GitHub, etc.
- support for version control, issues, etc.
- example:
 - <https://github.com/rdicosmo/parmap>
 - <https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/>

Where is the source code?

Collaborative development platforms (aka "forges")

- BitBucket, GitLab(.com), GitHub, etc.
- support for version control, issues, etc.
- example:
 - <https://github.com/rdicosmo/parmap>
 - <https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/>

Distribution platforms

- CTAN, CRAN, PyPi, Debian, etc.
- example: <https://ctan.org/pkg/biblatex-software>

Where is the source code?

Collaborative development platforms (aka "forges")

- BitBucket, GitLab(.com), GitHub, etc.
- support for version control, issues, etc.
- example:
 - <https://github.com/rdicosmo/parmap>
 - <https://gitlab.inria.fr/gt-sw-citation/bibtex-sw-entry/>

Distribution platforms

- CTAN, CRAN, PyPi, Debian, etc.
- example: <https://ctan.org/pkg/biblatex-software>

Archives

- Software Heritage
- example: [archived version of biblatex-software](#)

Forges are *not* archives!

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)

Forges are *not* archives!

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)

2019: big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250.000 repositories (including research software)

Forges are *not* archives!

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)

2019: big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250.000 repositories (including research software)

2021: ... in Academia too

- october 2021: Inria's old gforge is unplugged
 - **breaks the build chain** of the OCaml package manager (Opam)

Forges are *not* archives!

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)

2019: big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250.000 repositories (including research software)

2021: ... in Academia too

- october 2021: Inria's old gforge is unplugged
 - **breaks the build chain** of the OCaml package manager (Opam)

Bottomline

we need a universal archive of software source code:

Forges are *not* archives!

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)

2019: big bad news keep coming in

- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250.000 repositories (including research software)

2021: ... in Academia too

- october 2021: Inria's old gforge is unplugged
 - **breaks the build chain** of the OCaml package manager (Opam)

Bottomline

we need a universal archive of software source code: now we have one!



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



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



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 all software
source code



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 all software
source code

Research infrastructure



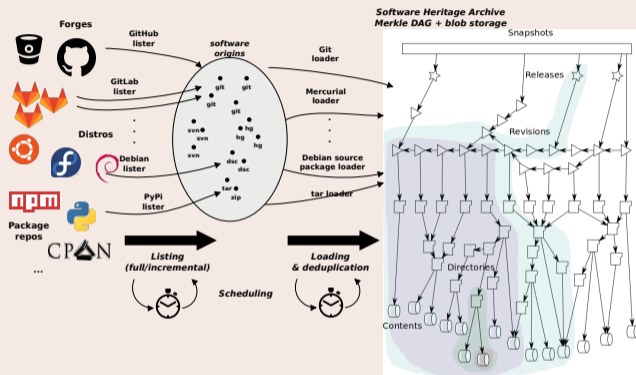
enable analysis of all
software source code

Addressing the A(rchive)

Universal source code archive

not only research

(11B+ files, 160M+ projects)



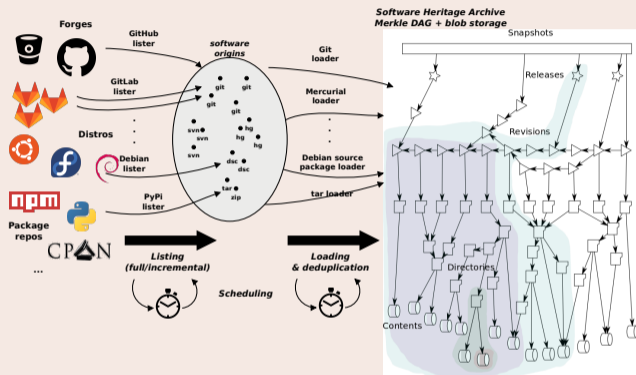
- your research software *is likely there already!*

Addressing the A(rchive)

Universal source code archive

not only research

(11B+ files, 160M+ projects)



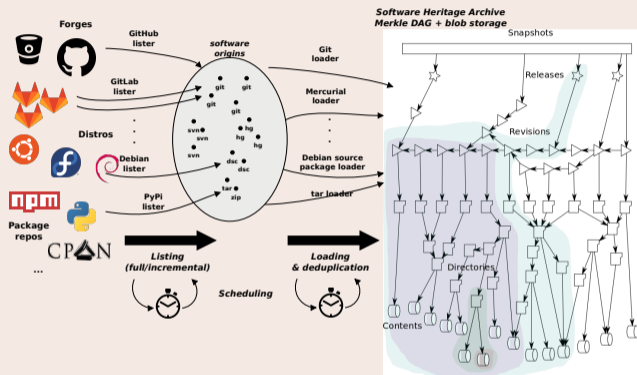
- your research software *is likely there already!*
- anyone can trigger archival with save.softwareheritage.org

Addressing the A(rchive)

Universal source code archive

not only research

(11B+ files, 160M+ projects)



- your research software *is likely there already!*
- anyone can trigger archival with save.softwareheritage.org
- selected partners can push to the archive via deposit.softwareheritage.org

Saving 250.000 endangered repositories...

- summer 2019: BitBucket announce Mercurial VCS phase out
- fall 2019: Software Heritage teams up with Octopus (funded by NLNet, thanks!)
- july 2020: BitBucket erases *250.000* repositories
- august 2020: bitbucket-archive.softwareheritage.org is live

Saving 250.000 endangered repositories...

- summer 2019: BitBucket announce Mercurial VCS phase out
- fall 2019: Software Heritage teams up with Octopus (funded by NLNet, thanks!)
- july 2020: BitBucket erases 250.000 repositories
- august 2020: bitbucket-archive.softwareheritage.org is live

... preserving the web of knowledge

(original tweet [is here](#))



Gabriel Altay
@gabrielaltay

Just realized [@Bitbucket](#) disabled all mercurial repositories when the [@asclnet](#) informed me that a link associated with an old paper of mine was down. Thought all was lost, but someone archived all the repos! very classy move by [@octopus_net](#) and [@SWHeritage](#).

[Traduire le Tweet](#)

1:48 AM · 31 août 2020 · Twitter Web App

Bottomline

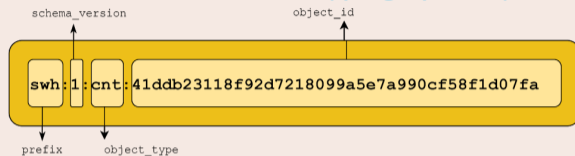
explicit deposit is important, ...

... and we must promote it...

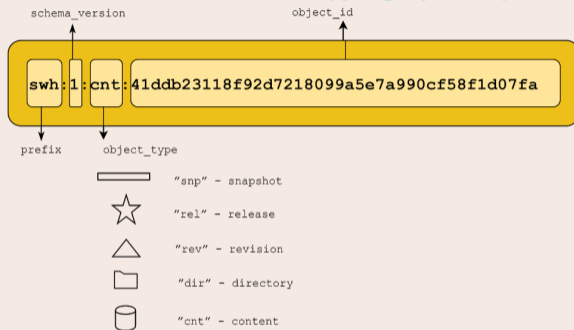
... but will never be enough.

(think also of all software dependencies!)

Software Heritage Identifiers (SWHID)

[link to full docs](#)20+B **intrinsic, decentralised, cryptographically strong identifiers, SWHIDs**

Software Heritage Identifiers (SWHID)

[link to full docs](#)20+B **intrinsic, decentralised, cryptographically strong identifiers, SWHIDs**

Software Heritage Identifiers (SWHID)

[link to full docs](#)20+B **intrinsic, decentralised, cryptographically strong identifiers, SWHIDs**

Software Heritage Identifiers (SWHID)

[link to full docs](#)20+B **intrinsic, decentralised, cryptographically strong identifiers, SWHIDs**Emerging standard : Linux Foundation [SPDX 2.2](#); IANA registered; WikiData [P6138](#)

Software Heritage Identifiers (SWHID)

[link to full docs](#)20+B **intrinsic, decentralised, cryptographically strong identifiers, SWHIDs**Emerging standard : Linux Foundation [SPDX 2.2](#); IANA registered; WikiData [P6138](#)Full fledged *source code references* for reproducibilityExamples: [Apollo 11 AGC excerpt](#), [Quake III rsqrt](#); Guidelines available, see [ICMS 2020](#)

Describe

- Collect *intrinsic metadata*
- Contributed the [Codemeta generator](#)

CodeMeta generator

Most fields are optional. Mandatory fields will be highlighted when generating Codemeta.

The software itself

Name

My Software
the software title

Description

My Software computes ephemerides and orbit propagation. It has been developed from early '80.

Creation date

YYYY-MM-DD

First release date

YYYY-MM-DD

Describe

- Collect *intrinsic metadata*
- Contributed the [Codemeta generator](#)

CodeMeta generator

Most fields are optional. Mandatory fields will be highlighted when generating Codemeta.

The software itself

Name

My Software
the software title

Description

My Software computes ephemerides and orbit propagation. It has been developed from early '80.

Creation date

YYYY-MM-DD

First release date

YYYY-MM-DD

Cite/Credit

- Contributed *software citation style*
[biblatex-software, v 1.2-2 now on CTAN](#)



- 1 Introduction
- 2 Open Science
- 3 Building the software pillar of Open Science: assessing the needs
- 4 Phase 1: focus on ARDC and infrastructures
- 5 Demo time!
- 6 Phase 2: broader policy issues



- Browse [the archive](#)
- [Trigger archival](#) of your preferred software in a breeze
- Get and use SWHIDs ([full specification available online](#))
- The [Apollo 11 AGC source code example](#)
- Cite software [with the biblatex-software style](#) from CTAN
- Example use in a research article: compare Fig. 1 and conclusions
 - in [the 2012 version](#)
 - in [the updated version](#) using SWHIDs and Software Heritage
- Example in a journal: [an article from IPOL](#)
- [Curated deposit in SWH via HAL](#), see for example: [LinBox](#), [SLALOM](#), [Givaro](#), [NS2DDV](#), [SumGra](#), [Coq proof](#), ...
- Rescue landmark legacy software, see the [SWHAP process with UNESCO](#)

Growing adoption of SWH in Academia (selection)

HAL software curated deposit workflow

Curated Archiving of Research Software Artifacts

International Journal of Digital Curation, 2020

Reference archive for swmath.org



See *code* links, e.g.

[SemiPar package](#)

Growing adoption of SWH in Academia (selection)

HAL software curated deposit workflow

Curated Archiving of Research Software Artifacts

International Journal of Digital Curation, 2020

Reference archive for swmath.org



See *code* links, e.g.

[SemiPar package](#)

IPOL (image processing)



- archive (deposit)
- reference
- [BibLaTeX](#)

eLife (life sciences)



- archive (save code now)
- reference

JTCAM (Mechanics)

- [instructions for authors](#)
- [biblatex-software](#) in journal \LaTeX class

Growing adoption of SWH in Academia (selection)

HAL software curated deposit workflow

Curated Archiving of Research Software Artifacts

International Journal of Digital Curation, 2020

Reference archive for swmath.org



See *code* links, e.g.

[SemiPar package](#)

IPOL (image processing)



- archive (deposit)
- reference
- [BibLaTeX](#)

eLife (life sciences)



- archive (save code now)
- reference

JTCAM (Mechanics)

- [instructions for authors](#)
- [biblatex-software](#) in journal \LaTeX class

Policy: France



*National Plan
Open Science*

Policy: Europe



EOSC SIRS report

- SWHIDs
- archive

Guidelines



Software Heritage

- 1 Prepare your public repository (academic, software & scientific files)
- 2 Save your code (<http://osdn.sourceforge.net.org>)
- 3 Reference your work (full repository, specific version or code fragments)

- [summary](#)
- [ICMS 2020](#)

Sharing the vision



United Nations
Educational, Scientific and
Cultural Organization



And many more ...

www.softwareheritage.org/support/testimonials

Sharing the vision



United Nations
Educational, Scientific and
Cultural Organization



And many more ...

www.softwareheritage.org/support/testimonials

Donors, members, sponsors



Platinum sponsors



Gold sponsors



Silver sponsors



Bronze sponsors



Call to action on ARDC: let's foster adoption!

Train students and colleagues to [archive and reference relevant source code](#)

- full details in the [ICMS 2020](#) article
- short operational [HOWTO online](#)

Call to action on ARDC: let's foster adoption!

Train students and colleagues to [archive and reference relevant source code](#)

- full details in the [ICMS 2020](#) article
- short operational [HOWTO online](#)

Engage conferences, journals, learned societies to use Software Heritage and SWHIDs

APIs for [save code now](#) and [deposit](#) are available to integrate with

- Research Articles
- Artifact Evaluation Committees
- Badging initiatives

Call to action on ARDC: let's foster adoption!

Train students and colleagues to [archive and reference relevant source code](#)

- full details in the [ICMS 2020](#) article
- short operational [HOWTO online](#)

Engage conferences, journals, learned societies to use Software Heritage and SWHIDs

APIs for [save code now](#) and [deposit](#) are available to integrate with

- Research Articles
- Artifact Evaluation Committees
- Badging initiatives

Help grow and structure the community

- Promote the [ambassador program](#)
- Encourage our institutions to
 - include Software Heritage in their Open Science policy
 - become [member/sponsor](#)
 - build a Software Heritage mirror (see ENEA)

- 1 Introduction
- 2 Open Science
- 3 Building the software pillar of Open Science: assessing the needs
- 4 Phase 1: focus on ARDC and infrastructures
- 5 Demo time!
- 6 Phase 2: broader policy issues



Recall: beyond ARDC

Policy for dissemination and reuse

- open source research software
- revisit technology transfer and industry collaboration

Recall: beyond ARDC

Policy for dissemination and reuse

- open source research software
- revisit technology transfer and industry collaboration

Framework for evaluation and recognition

- make software development count in a career...
 - not the case in many countries (e.g. Italy)

Recall: beyond ARDC

Policy for dissemination and reuse

- open source research software
- revisit technology transfer and industry collaboration

Framework for evaluation and recognition

- make software development count in a career...
 - not the case in many countries (e.g. Italy)
- ... but avoid the number games
 - counting citations and commits *is not the silver bullet*
 - acknowledge the complexity of the task

Recall: beyond ARDC

Policy for dissemination and reuse

- open source research software
- revisit technology transfer and industry collaboration

Framework for evaluation and recognition

- make software development count in a career...
 - not the case in many countries (e.g. Italy)
- ... but avoid the number games
 - counting citations and commits *is not the silver bullet*
 - acknowledge the complexity of the task

Sustainability

technical improve quality of *key* research software

financial make research software as easy to fund as buying a license (somewhat similar issues with Open Access)

Recall: beyond ARDC

Policy for dissemination and reuse

- open source research software
- revisit technology transfer and industry collaboration

Framework for evaluation and recognition

- make software development count in a career...
 - not the case in many countries (e.g. Italy)
- ... but avoid the number games
 - counting citations and commits *is not the silver bullet*
 - acknowledge the complexity of the task

Sustainability

technical improve quality of *key* research software

financial make research software as easy to fund as buying a license (somewhat similar issues with Open Access)

Recall: beyond ARDC

Policy for dissemination and reuse

- open source research software
- revisit technology transfer and industry collaboration

Framework for evaluation and recognition

- make software development count in a career...
 - not the case in many countries (e.g. Italy)
- ... but avoid the number games
 - counting citations and commits *is not the silver bullet*
 - acknowledge the complexity of the task

Sustainability

technical improve quality of *key* research software

financial make research software as easy to fund as buying a license (somewhat similar issues with Open Access)

The UNESCO recommendations for Open Science, 2018-2021



United Nations
Educational, Scientific and
Cultural Organization

Ref.: CL/4363

Subject: **Draft text of the UNESCO Recommendation on Open Science**

Madam/Sir,

At its 40th session in November 2019, the UNESCO General Conference decided to elaborate a draft Recommendation on Open Science.

This was a major decision, which has since mobilized the entire Organization and all of its Member States in the development of this new standard-setting instrument.

After two years of joint work, this process is now entering its final phase, following the consensus reached on the draft text during the intergovernmental meeting of experts held from 6 to 11 May 2021.

I have the pleasure to submit to you this draft recommendation, which will be put forward for adoption at our next General Conference in November 2021.

The definitions and principles that it contains constitute a common – and currently unprecedented – framework to support scientific cooperation and make science more transparent, more accessible, more equitable and more inclusive.

For any further information, Shamila Nair-Bedouelle, Assistant Director-General for Natural Sciences, is at your disposal at the following email address: open-science@unesco.org

Thanking you for your commitment, please accept, Madam/Sir, the assurances of my highest consideration.

Handwritten signature of Audrey Azoulay in black ink.

Audrey Azoulay
Director-General

Enclosure: 1

1. Draft text of the UNESCO Recommendation on Open Science

cc: Permanent Delegations to UNESCO
National Commissions for UNESCO

1, place de France
75116 Paris CEDEX 07, France
Tel: +33 (0)7 53 42 35 10
www.unesco.org

To Ministers responsible for relations with UNESCO

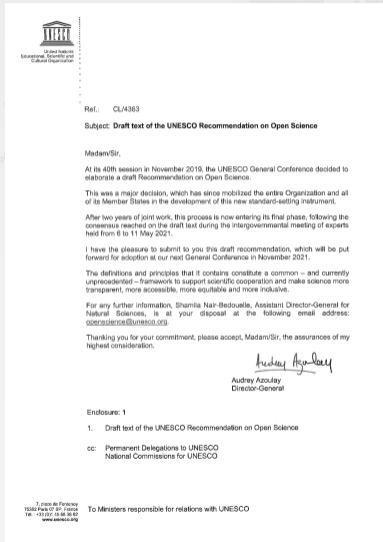
Selection from [the draft recommendations](#)

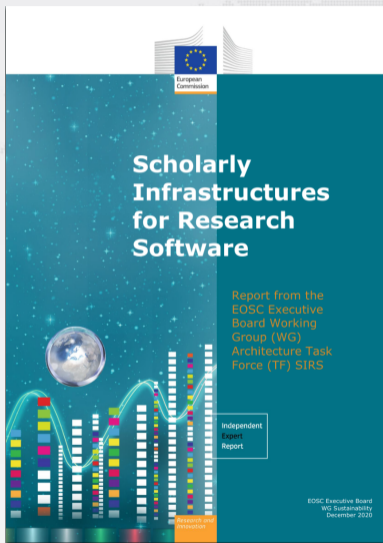
- Open Source for Open Science

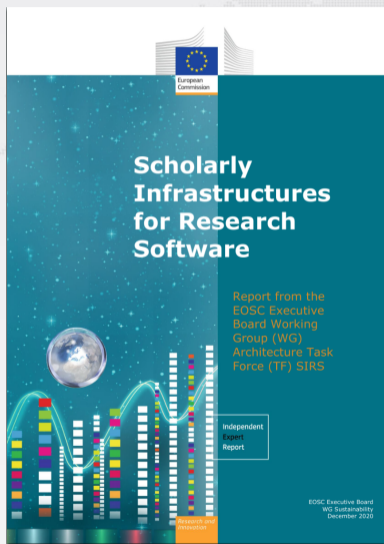
"The source code must be included in the software release and made available on openly accessible repositories and the chosen license must allow modifications, derivative works and sharing under equal or compatible open terms and conditions"

- Infrastructures

"Open science infrastructures should be organized and financed upon an essentially not-for-profit and long-term vision, which enhance open science practices and guarantee permanent and unrestricted access to ail, to thé largest extent possible."





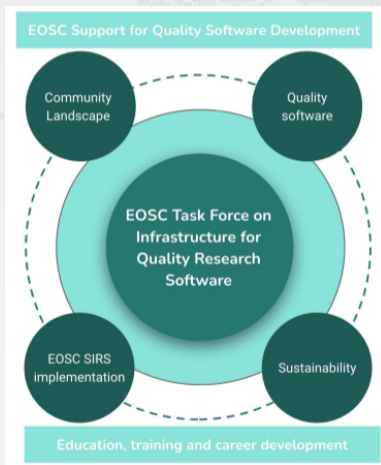


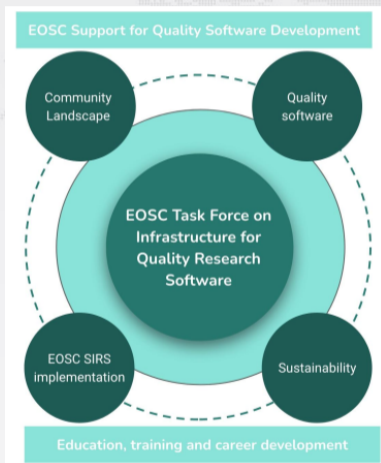
Important *policy tool* in Open Science (Dec 2020)

- 9 infrastructures
 - 3 archives
 - 3 open access publishers
 - 3 aggregators
- recommendations
 - archive in Software Heritage, use SWHID
 - open non profit
 - default to open source for research software

"all research software should be made available under an Open Source license by default, and all deviations from this default practice should be properly motivated"

See <https://doi.org/10.2777/28598>





Ongoing action in the EOSC

Task force on infrastructures for quality research software

- Foster the development and deployment of tools and services that allow researchers to properly archive, reference, describe with proper metadata, share and reuse research software.
- Improve the quality of research software, both from the technical and organizational point of view ...
- Increase recognition to software developers and maintainers of research software ...

See [the charter of the task force](#).

French National plan for Open Science, 2021-2024





2nd National Plan for Open Science (6/7/2021)

Open and promote research software source code

- actions (selection)
 - charter for research software policy
 - recognize software development (see [announcement of the 2021 prize](#))
 - coordinate communities of practice
 - build a connected ecosystem of research outputs
- recommendations (selection)
 - archive in Software Heritage
 - standardise and use SWHID
 - build a national catalog of research software
 - leverage ADAC network

See [official announcement](#)

Call to action: let's engage with policy makers (it may be us!)

Institutional representation

we need an (open source) software VP in

- universities
- ministries
- governments

Call to action: let's engage with policy makers (it may be us!)

Institutional representation

we need an (open source) software VP in

- universities
- ministries
- governments

Funding for infrastructures

push for funding instruments adapted to digital infrastructures (e.g. ESFRI):

- cost of human resources is *predominant*
- *much shorter* time frame

Call to action: let's engage with policy makers (it may be us!)

Institutional representation

we need an (open source) software VP in

- universities
- ministries
- governments

Funding for infrastructures

push for funding instruments adapted to digital infrastructures (e.g. ESFRI):

- cost of human resources is *predominant*
- *much shorter* time frame

Set the default to open: pass the message

publicly funded research software should be open source

exceptions must be justified

Call to action: let's engage with policy makers (it may be us!)

Institutional representation

we need an (open source) software VP in

- universities
- ministries
- governments

Funding for infrastructures

push for funding instruments adapted to digital infrastructures (e.g. ESFRI):

- cost of human resources is *predominant*
- *much shorter* time frame

Set the default to open: pass the message

publicly funded research software should be open source

exceptions must be justified






Career evaluation and incentives

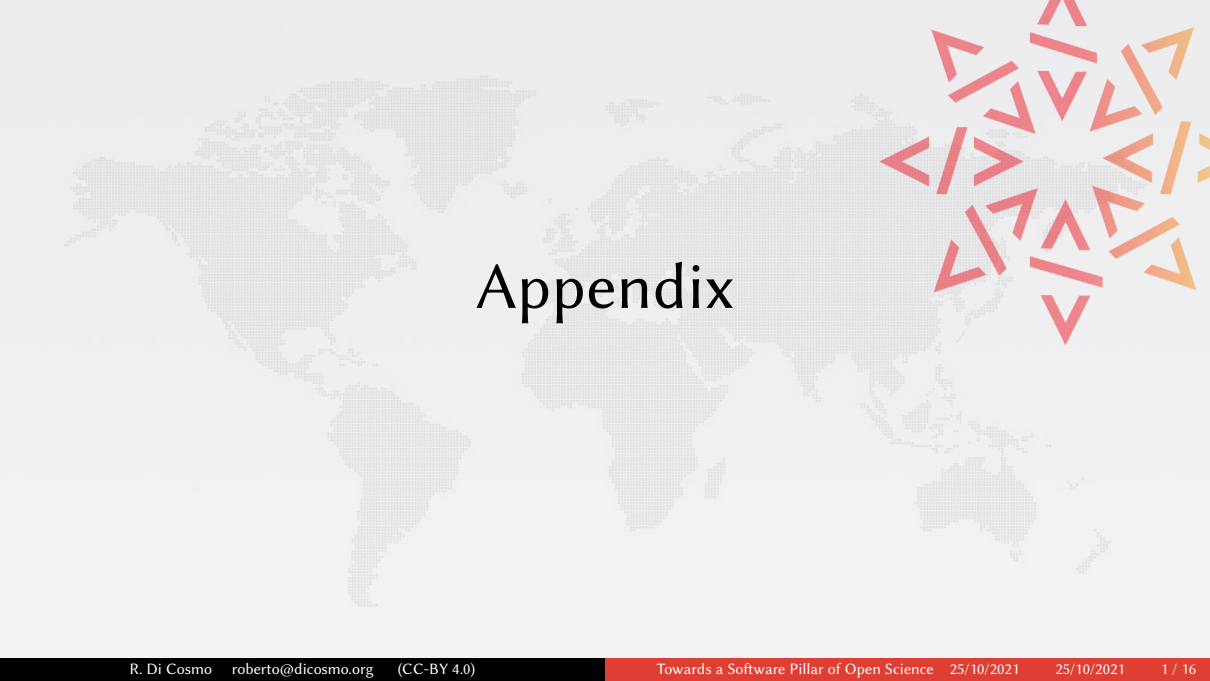
- recognize *quality* software development
 - see e.g. [the 2021 Inria guidelines](#) (in french) and [this CiSE 2020 article](#) (in english)
- keep the human in the loop, avoid number games

it's a long road, but together we can make it

Questions?

References

-  UNESCO, *Draft recommendations on Open Science* 2021, ([online](#))
-  French Ministry of Research, *Second National Plan for Open Science* 2021, ([online](#))
-  EOSC SIRS Task Force, *Scholarly Infrastructures for Research Software* 2020, Publications office of the European Commission, ([10.2777/28598](#))
-  R. Di Cosmo, *Archiving and Referencing Source Code with Software Heritage* International Conference on Mathematical Software 2020 ([10.1007/978-3-030-52200-1_36](#))
-  J.F. Abramatic, R. Di Cosmo, S. Zacchiroli, *Building the Universal Archive of Source Code* CACM, October 2018 ([10.1145/3183558](#))



Appendix

- 
- 7 FAIR
 - 8 SWHIDs
 - 9 Revolutionary infrastructure, scientific challenges
 - 10 All the source code
 - 11 Milestones, policy
 - 12 Public code, mirrors

B_BLOCK

What about FAIR? (Findable, Accessible, Interoperable, Reusable)

FAIR data principles *for data*

in a nutshell: metadata, metadata, metadata all over the place (makes sense for data)

What about FAIR? (Findable, Accessible, Interoperable, Reusable)

FAIR data principles *for data*

in a nutshell: metadata, metadata, metadata all over the place (makes sense for data)

But software is *not data* ...

the terms *interoperability* and *reusability* have precise technical meaning for software, and *differ significantly* from what is intended by the I and R of FAIR;

- see the entries for [software interoperability](#) and [software reusability](#)
- it is *very difficult* to achieve these properties even for commercial software developed by multi billion dollars corporations

What about FAIR? (Findable, Accessible, Interoperable, Reusable)

FAIR data principles *for data*

in a nutshell: metadata, metadata, metadata all over the place (makes sense for data)

But software is *not data* ...

the terms *interoperability* and *reusability* have precise technical meaning for software, and *differ significantly* from what is intended by the I and R of FAIR;

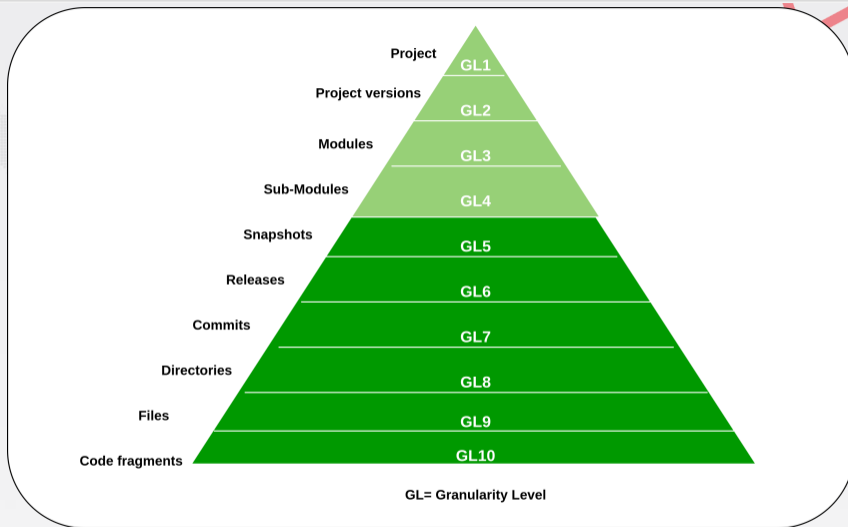
- see the entries for [software interoperability](#) and [software reusability](#)
- it is *very difficult* to achieve these properties even for commercial software developed by multi billion dollars corporations

FAIR for software is a distraction

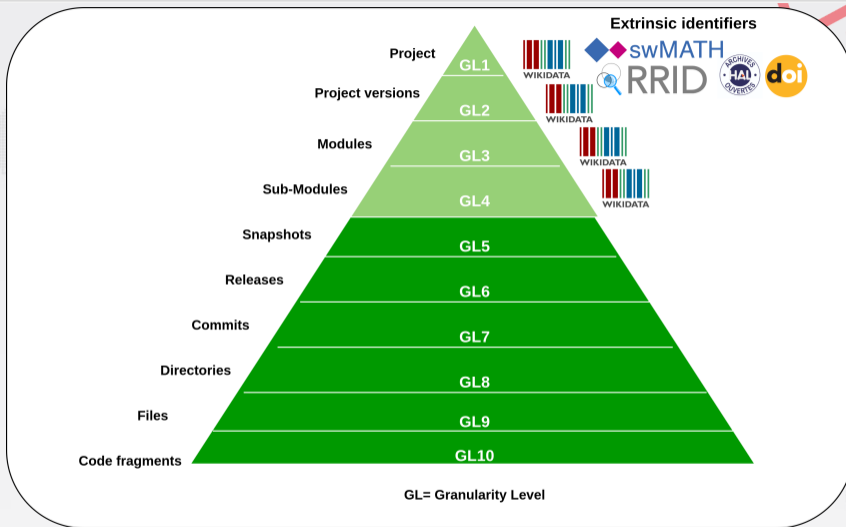
let's focus on the real issues at stake: ARDC a good starting point

- 
- 7 FAIR
 - 8 SWHIDs
 - 9 Revolutionary infrastructure, scientific challenges
 - 10 All the source code
 - 11 Milestones, policy
 - 12 Public code, mirrors

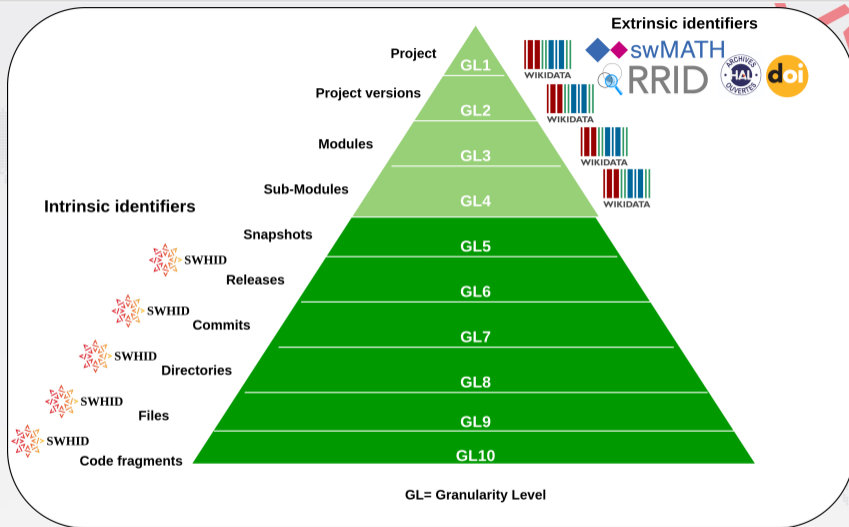
B_BLOCK



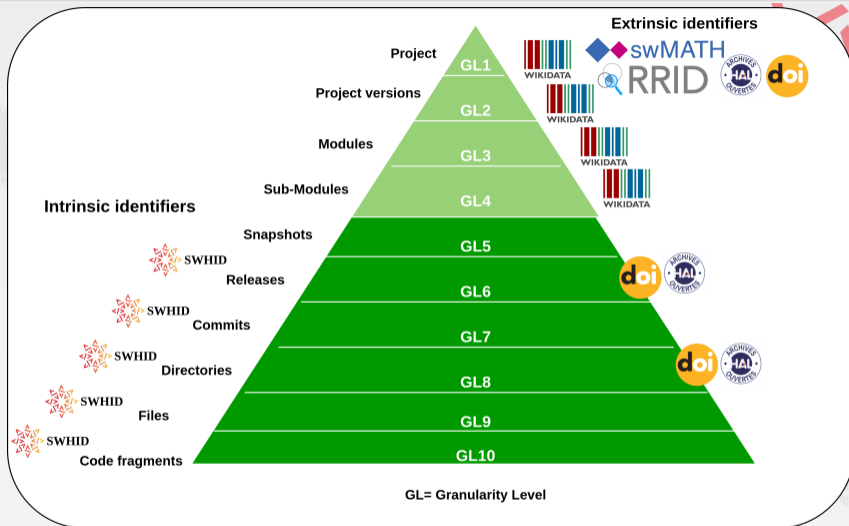
Top concept layers vs. bottom artifact layers



Extrinsic identifiers are key for the concept layers



Intrinsic identifiers are key for the artifact layers



In some cases, extrinsic identifiers can be added too

Extrinsic and Intrinsic identifiers in a nutshell

Extrinsic identifiers: no *per se* relation with the designated Object

A *register* keeps the correspondence between the identifier and the object

pre-internet era passport number, social security number, ISBN, ISSN, etc.

internet era DOI, Handle, Ark, PURLs, RRID, etc.

Extrinsic and Intrinsic identifiers in a nutshell

Extrinsic identifiers: no *per se* relation with the designated Object

A *register* keeps the correspondence between the identifier and the object

pre-internet era passport number, social security number, ISBN, ISSN, etc.

internet era DOI, Handle, Ark, PURLs, RRID, etc.

Intrinsic identifiers: derived from the designated Object

No register needed to keep the correspondence between the identifier and the object

pre-internet era musical notation, chemical notation (*NaCl* is table salt)

Extrinsic and Intrinsic identifiers in a nutshell

Extrinsic identifiers: no *per se* relation with the designated Object

A *register* keeps the correspondence between the identifier and the object

pre-internet era passport number, social security number, ISBN, ISSN, etc.

internet era DOI, Handle, Ark, PURLs, RRID, etc.

Intrinsic identifiers: derived from the designated Object

No register needed to keep the correspondence between the identifier and the object

pre-internet era musical notation, chemical notation (*NaCl* is table salt)

internet era cryptographic hashes for distributed software development, Bitcoin

Extrinsic and Intrinsic identifiers in a nutshell

Extrinsic identifiers: no *per se* relation with the designated Object

A *register* keeps the correspondence between the identifier and the object

pre-internet era passport number, social security number, ISBN, ISSN, etc.

internet era DOI, Handle, Ark, PURLs, RRID, etc.

Intrinsic identifiers: derived from the designated Object

No register needed to keep the correspondence between the identifier and the object

pre-internet era musical notation, chemical notation (*NaCl* is table salt)

internet era cryptographic hashes for distributed software development, Bitcoin

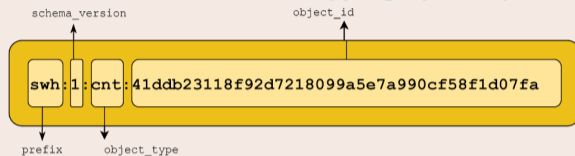
more in [this dedicated blog post](#) (with pointers to literature)

Meet the SWHID intrinsic identifiers

Software Heritage Identifiers (SWHID)

[link to full docs](#)

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

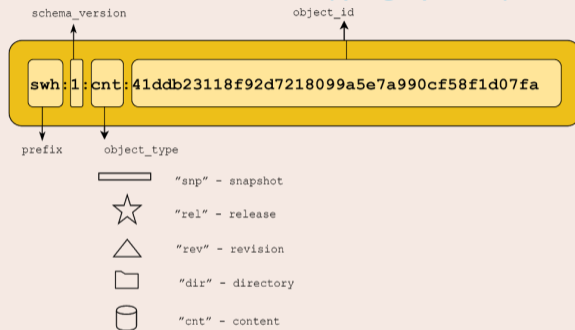


Meet the SWHID intrinsic identifiers

Software Heritage Identifiers (SWHID)

[link to full docs](#)

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



Meet the SWHID intrinsic identifiers

Software Heritage Identifiers (SWHID)

[link to full docs](#)

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



Meet the SWHID intrinsic identifiers

Software Heritage Identifiers (SWHID)

[link to full docs](#)

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



Emerging standard : Linux Foundation [SPDX 2.2](#); IANA registered; WikiData [P6138](#)

Meet the SWHID intrinsic identifiers

Software Heritage Identifiers (SWHID)

[link to full docs](#)

20+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](#)

- 
- 7 FAIR
 - 8 SWHIDs
 - 9 Revolutionary infrastructure, scientific challenges
 - 10 All the source code
 - 11 Milestones, policy
 - 12 Public code, mirrors

B_BLOCK

Summing up: a revolutionary infrastructure *designed for source code*



Software Heritage

www.softwareheritage.org

global source code archive

Library of Alexandria of source code



- harvest *all* software source code
- *on demand* harvesting and *curated deposit*

universal intrinsic identifiers

SWHID standard is independent of version control systems

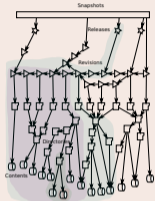
uniform data model, *full graph* of development history

enables large scale, big code research

infrastructure for Open Science

base layer for software source code in the *Open Science architecture*

The *graph* of Software Development

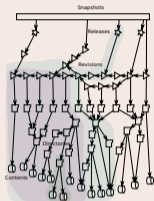


All software development
in a **single graph** ...



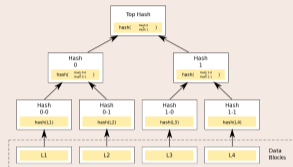
A revolutionary infrastructure *designed for software source code*

The *graph* of Software Development



All software development
in a **single graph** ...

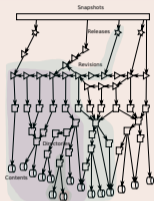
The *blockchain* of Software Development



... a single
Merkle graph!

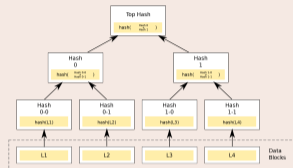
A revolutionary infrastructure *designed for software source code*

The *graph* of Software Development



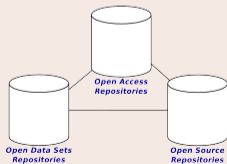
All software development
in a **single graph** ...

The *blockchain* of Software Development



... a single
Merkle graph!

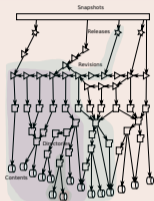
A *pillar* of Open Science



Reference **archive** of
Research Software

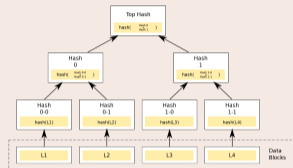
A revolutionary infrastructure *designed for software source code*

The *graph* of Software Development



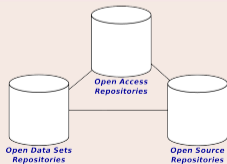
All software development
in a **single graph** ...

The *blockchain* of Software Development



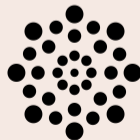
... a single
Merkle graph!

A *pillar* of Open Science



Reference **archive** of
Research Software

Reference platform for *Big Code*



A **single, uniform** data structure

A challenging scientific and technical undertaking

A novel, large infrastructure

- object storage [with peculiar workload](#)
- gigantic Merkle graph
- counting tens of billions of objects ([reuse P. Flajolet's seminal work](#))
- and much more: see [the 2021 technical roadmap](#)

A challenging scientific and technical undertaking

A novel, large infrastructure

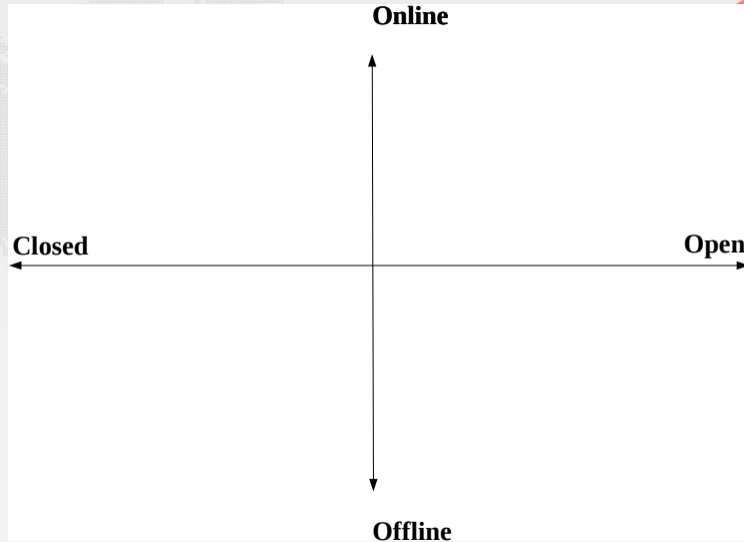
- object storage [with peculiar workload](#)
- gigantic Merkle graph
- counting tens of billions of objects ([reuse P. Flajolet's seminal work](#))
- and much more: see [the 2021 technical roadmap](#)

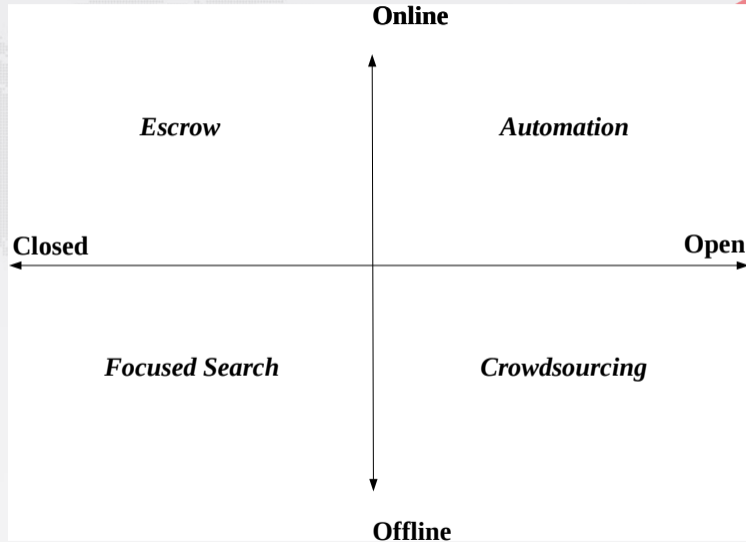
First datasets are available for Big Code analysis

- full graph of software development (~20Bn nodes, ~200Bn edges) see Pietri, Spinellis, Zacchiroli, MSR 2019
<https://dx.doi.org/10.1109/MSR.2019.00030>
- MSR 2020 mining competition see <https://2020.msrconf.org/track/msr-2020-mining-challenge#Call-for-Papers>

- 
- 7 FAIR
 - 8 SWHIDs
 - 9 Revolutionary infrastructure, scientific challenges
 - 10 All the source code**
 - 11 Milestones, policy
 - 12 Public code, mirrors

B_BLOCK





- 
- 7 FAIR
 - 8 SWHIDs
 - 9 Revolutionary infrastructure, scientific challenges
 - 10 All the source code
 - 11 Milestones, policy
 - 12 Public code, mirrors

B_BLOCK

Summer 2015



The collection starts: first server, (very) early prototype

June 30th 2016



Public unveiling, with the first sponsors: Microsoft and DANS

Summer 2015



The collection starts: first server, (very) early prototype

June 30th 2016



Public unveiling, with the first sponsors: Microsoft and DANS

April 3rd 2017



Unesco - Inria agreement on software access and preservation.

June 7th 2018



Opening the archive to the world

Summer 2015



The collection starts: first server, (very) early prototype

June 30th 2016



Public unveiling, with the first sponsors: Microsoft and DANS

April 3rd 2017



Unesco - Inria agreement on software access and preservation.

June 7th 2018



Opening the archive to the world

December 7th 2018



Starting the mirror network

February 26th 2019



Publication of the expert meeting
**Paris Call on
Software Source Code**

"Upload filters": a threat to *all modern software development*

- developing platforms (GitHub, GitLab, Bitbucket, etc.)
- distribution platforms (Maven, Pypi, CRAN, CTAN, etc.)
- archives (Software Heritage)

"Upload filters": a threat to *all modern software development*

- developing platforms (GitHub, GitLab, Bitbucket, etc.)
- distribution platforms (Maven, Pypi, CRAN, CTAN, etc.)
- archives (Software Heritage)

We got an exclusion for

non for profit open source software developing and sharing platforms

"Upload filters": a threat to *all modern software development*

- developing platforms (GitHub, GitLab, Bitbucket, etc.)
- distribution platforms (Maven, Pypi, CRAN, CTAN, etc.)
- archives (Software Heritage)

We got an exclusion for

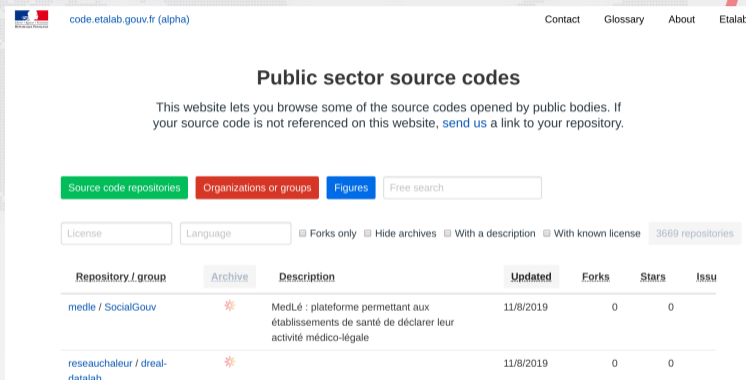
non for profit open source software developing and sharing platforms

Key role of Software Heritage

policy-maker awareness, essential insights for NGOs, government contacts

- 
- 7 FAIR
 - 8 SWHIDs
 - 9 Revolutionary infrastructure, scientific challenges
 - 10 All the source code
 - 11 Milestones, policy
 - 12 Public code, mirrors

B_BLOCK



The screenshot shows the website code.etalab.gouv.fr (alpha). The main heading is "Public sector source codes". Below this, a paragraph explains that the website lets users browse source codes opened by public bodies and provides a link to send a repository if not referenced. There are navigation buttons for "Source code repositories", "Organizations or groups", and "Figures", along with a "Free search" input field. Filter options include "License", "Language", "Forks only", "Hide archives", "With a description", "With known license", and a count of "3669 repositories". A table lists two repositories: "medle / SocialGouv" and "reseauchaleur / dreal-datalab", both marked as archived with a star icon. The table columns are Repository / group, Archive, Description, Updated, Forks, Stars, and Issu.

code.etalab.gouv.fr (alpha) Contact Glossary About Etalab

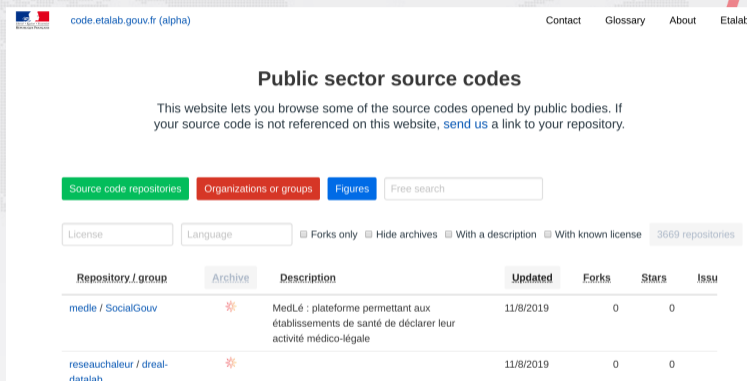
Public sector source codes

This website lets you browse some of the source codes opened by public bodies. If your source code is not referenced on this website, [send us](#) a link to your repository.

[Source code repositories](#) [Organizations or groups](#) [Figures](#)

Forks only Hide archives With a description With known license 3669 repositories

Repository / group	Archive	Description	Updated	Forks	Stars	Issu
medle / SocialGouv	★	MedLé : plateforme permettant aux établissements de santé de déclarer leur activité médico-légale	11/8/2019	0	0	
reseauchaleur / dreal-datalab	★		11/8/2019	0	0	



The screenshot shows the homepage of code.etalab.gouv.fr (alpha). The page features a navigation bar with links for Contact, Glossary, About, and Etalab. The main heading is "Public sector source codes", followed by a paragraph explaining the site's purpose: "This website lets you browse some of the source codes opened by public bodies. If your source code is not referenced on this website, [send us](#) a link to your repository." Below this, there are filter buttons for "Source code repositories" (green), "Organizations or groups" (red), and "Figures" (blue), along with a "Free search" input field. Further down, there are filters for "License" and "Language", and a list of checkboxes for "Forks only", "Hide archives", "With a description", and "With known license". A "3669 repositories" badge is also present. The main content area is a table with columns for "Repository / group", "Archive", "Description", "Updated", "Forks", "Stars", and "Issu". Two repositories are listed: "medle / SocialGov" and "reseauchaleur / dreal-datalab", both marked with a star icon.

code.etalab.gouv.fr (alpha) Contact Glossary About Etalab

Public sector source codes

This website lets you browse some of the source codes opened by public bodies. If your source code is not referenced on this website, [send us](#) a link to your repository.

[Source code repositories](#) [Organizations or groups](#) [Figures](#)

Forks only Hide archives With a description With known license 3669 repositories

Repository / group	Archive	Description	Updated	Forks	Stars	Issu
medle / SocialGov	★	MedLé : plateforme permettant aux établissements de santé de déclarer leur activité médico-légale	11/8/2019	0	0	
reseauchaleur / dreal-datalab	★		11/8/2019	0	0	

<https://code.etalab.gouv.fr>

Thomas Jefferson, February 18, 1791

...let us save what remains: not by vaults and locks which fence them from the public eye and use in consigning them to the waste of time, but by such a multiplication of copies, as shall place them beyond the reach of accident.

Thomas Jefferson, February 18, 1791

...let us save what remains: not by vaults and locks which fence them from the public eye and use in consigning them to the waste of time, but by such a multiplication of copies, as shall place them beyond the reach of accident.

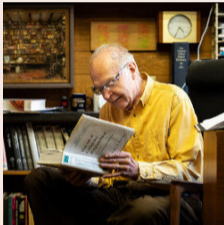
Welcoming ENEA



Italian National Agency for New Technologies,
Energy and Sustainable Economic Development

- first **institutional** mirror
- increased resilience
- **AI infrastructure** for researchers
- stepping stone to
an European joint effort

Communications of the ACM, February 2021



"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."

Let's Not Dumb Down the History of Computer Science

Donald E. Knuth, Len Shustek

<https://doi.org/10.1145/3442377>

Communications of the ACM, February 2021



"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."

Let's Not Dumb Down the History of Computer Science

Donald E. Knuth, Len Shustek

<https://doi.org/10.1145/3442377>

A unique opportunity

most of the creators are still here: we can talk to them!

but the clock is ticking...

Source code history for Security and Transparency

Where does reused software come from?



A word cloud containing various software-related terms and platforms. The most prominent words are 'Git Hub' in large purple letters, 'Sourceforge' in blue, and 'CRAN' in orange. Other visible words include 'Debian', 'Maven', 'Bitbucket', 'GoogleCode', 'Gitlab', 'CTAN', 'BerliOs', 'Adulact', 'Inria', 'Gitorious', and 'CPAN'.



Source code history for Security and Transparency

Where does reused software come from?



Do *you* know where it comes from?

- the software you ship
- the software you use
- the software you acquire
- the software that
 - has that bug
 - has that vulnerability

Source code history for Security and Transparency

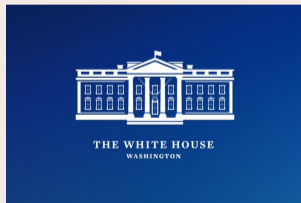
Where does reused software come from?



Do you know where it comes from?

- the software you ship
- the software you use
- the software you acquire
- the software that
 - has that bug
 - has that vulnerability

KYSW: Know Your SoftWare



Like KYC in banking, KYSW is now essential all over IT...

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