

# Software Heritage for Open Science and Open Source

## a revolutionary infrastructure

Roberto Di Cosmo  
ICSME 2022

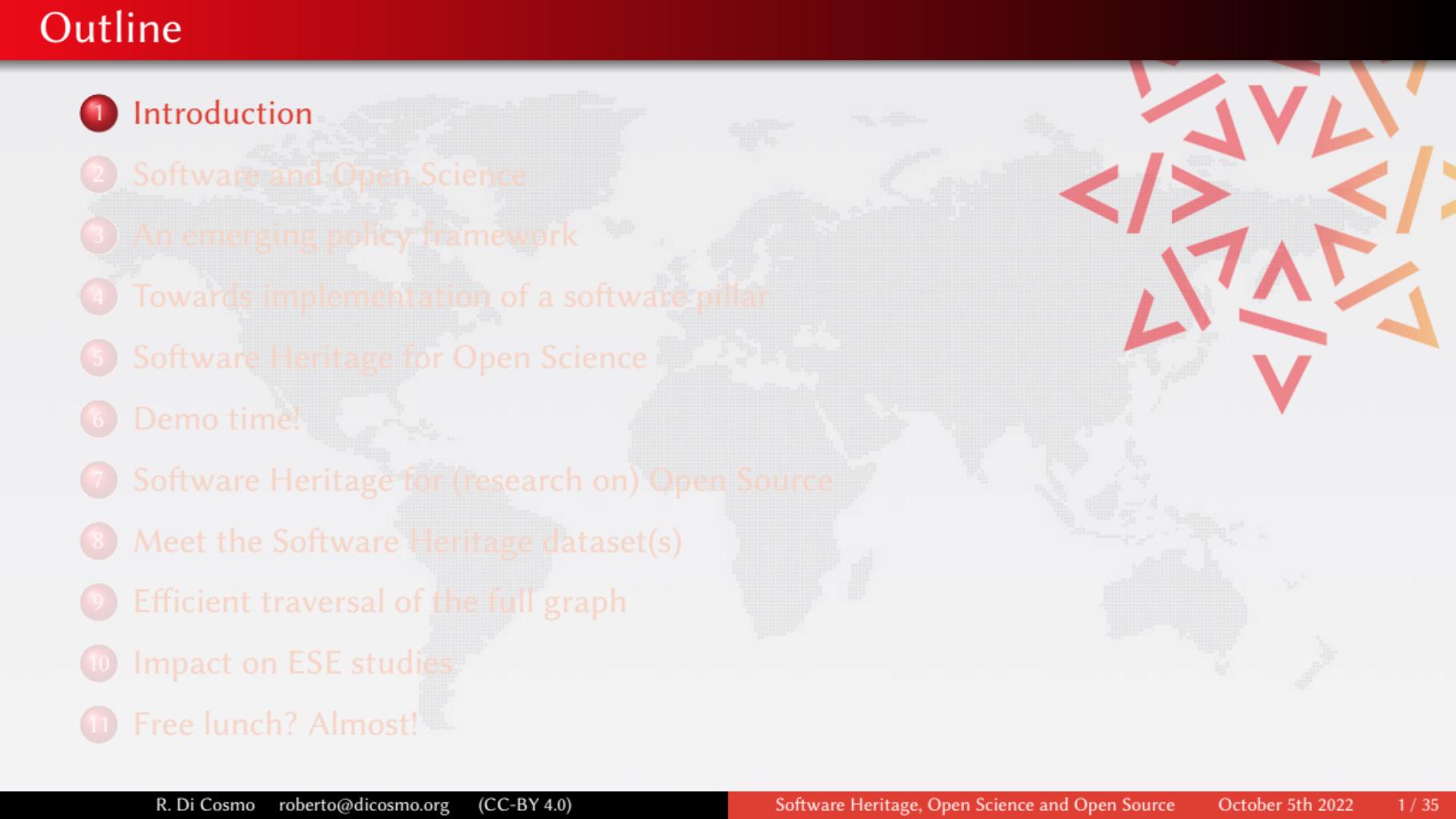
Director, Software Heritage  
Inria and Université de Paris Cité

October 5th 2022



Software Heritage  
THE GREAT LIBRARY OF SOURCE CODE

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

# Short Bio: Roberto Di Cosmo

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](http://www.mancoosi.org)

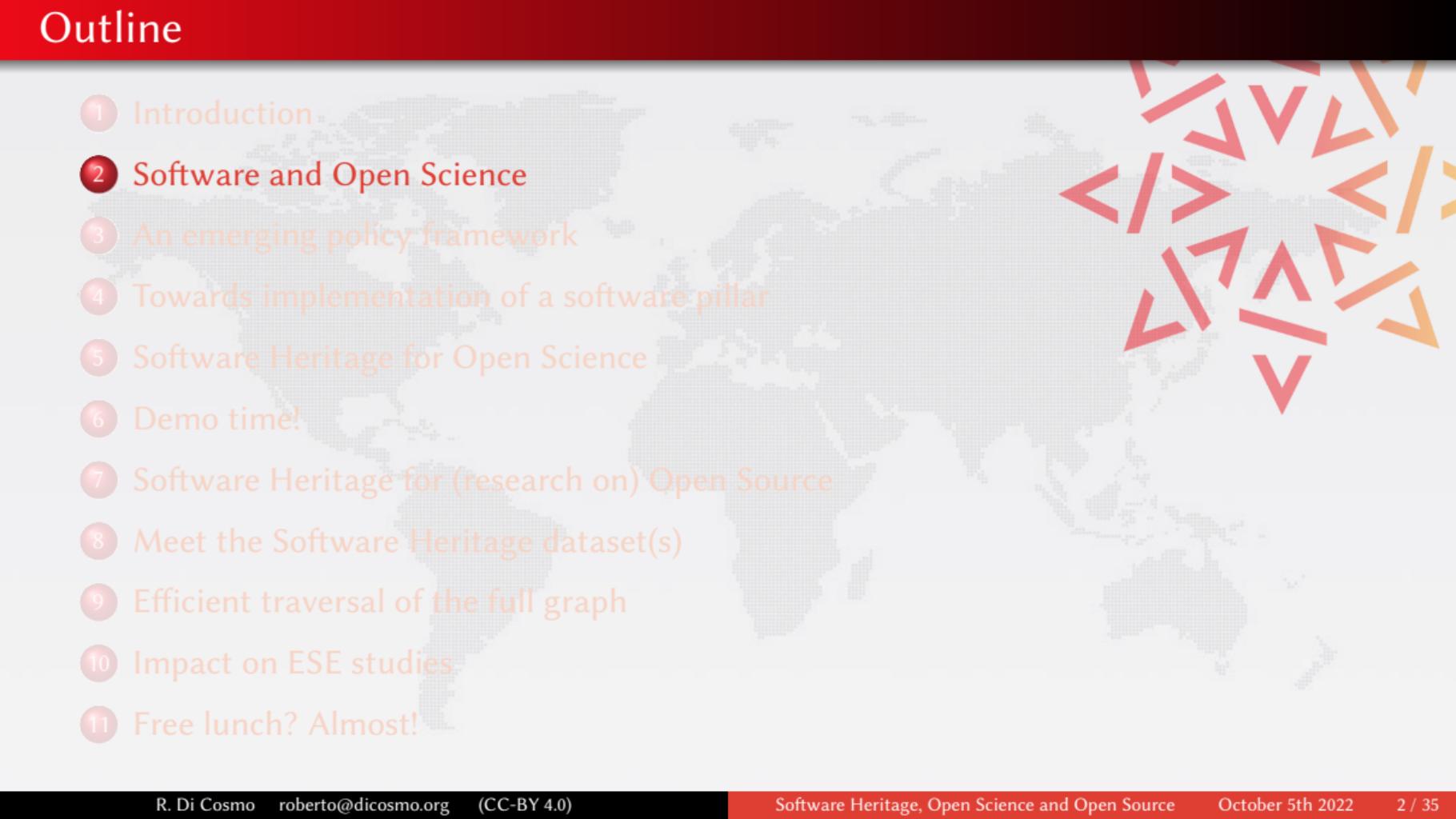
2010 *IRILL* [www.irill.org](http://www.irill.org)

2015 *Software Heritage* at INRIA

2018 *National Committee for Open Science*, France

2021 *EOSC Task Force on Infrastructures for Software*,  
European Union

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

# 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](#))

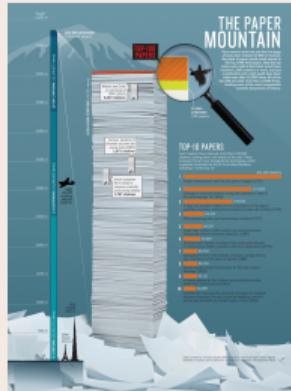
The COVID-19 crisis has also shown that cooperation at international level in research and innovation is more important than ever, including through *open access to data and results. No nation, no country can tackle any of these global challenges alone.*

Yuval Noah Harari (on COVID 19)

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

# Software: the third pillar of Open Science

Software powers modern research



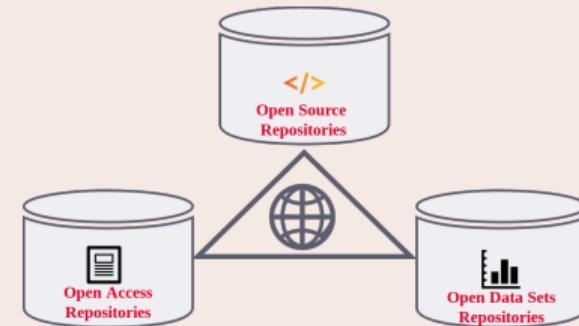
[...] software [...] essential in their fields.

*Top 100 papers (Nature, 2014)*

*Sometimes, if you dont have the software, you dont 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

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](#))

```
P63SPOT3    CA      BIT6          # IS THE LR ANTENNA IN POSITION 1 YET
EXTEND
RAND      CHAN33
EXTEND
BZF       P63SPOT4        # BRANCH IF ANTENNA ALREADY IN POSITION 1

CAF       CODE500        # ASTRONAUT: PLEASE CRANK THE
TC        BANKCALL       # SILLY THING AROUND
CADR     G0PERF1
TCF      GOTOPOOH       # TERMINATE
TCF      P63SPOT3        # PROCEED SEE IF HE'S LYING

P63SPOT4    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

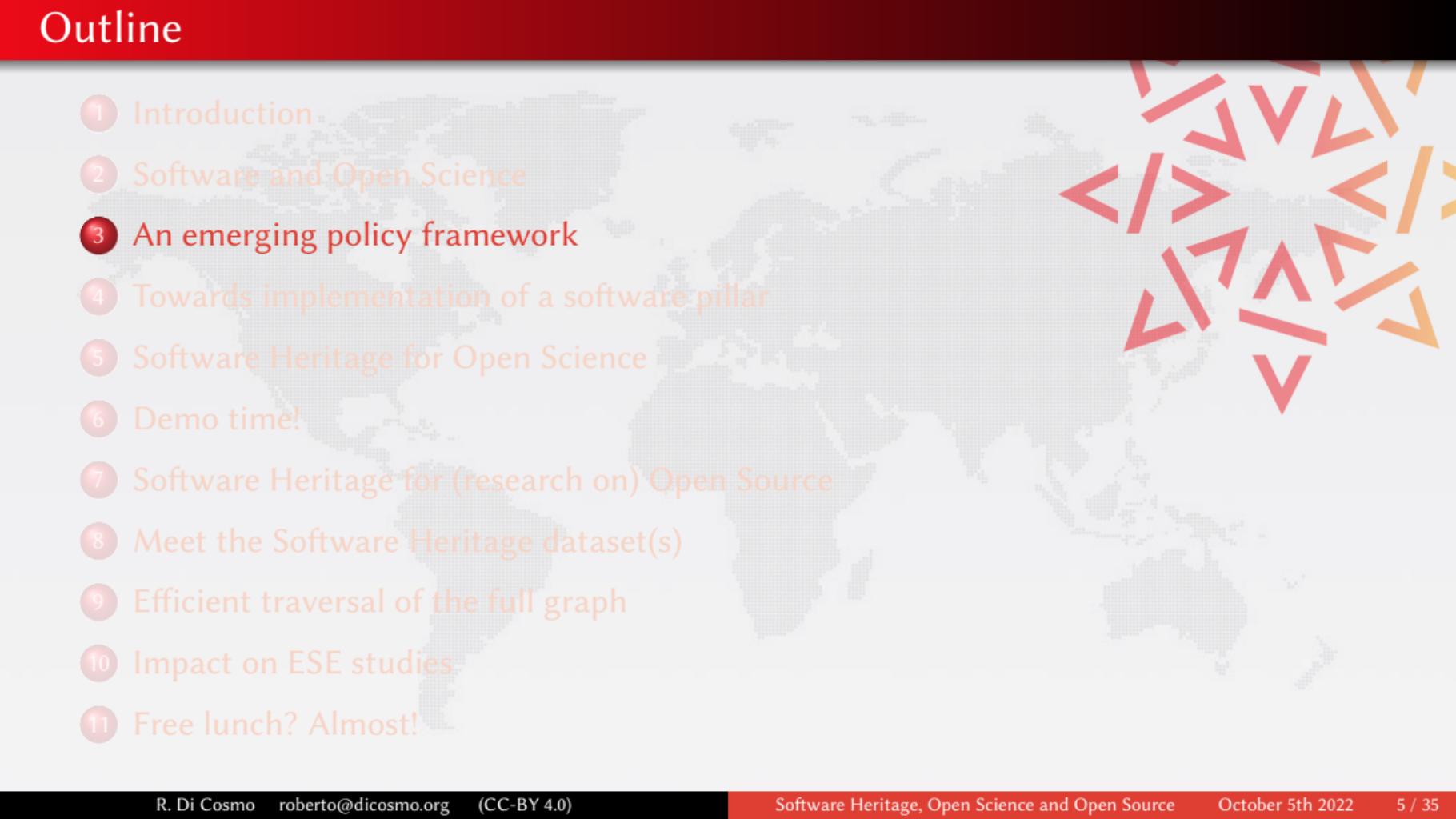
    return y;
}
```

Len Shustek, Computer History Museum

2006

*"Source code provides a view into the mind of the designer."*

# Outline

- 
- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

# International highlights

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



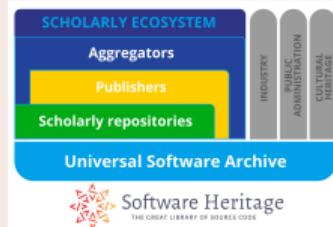
40 international experts call to “promote software development as a valuable research activity, and research software as a key enabler for Open Science/Open Research, [...] recognising in the careers of academics their contributions to high quality software development, in all their forms”

## UNESCO recommendations for Open Science, 2018-2021

“The source code must be included in the software release and [...] the license must allow modifications, derivative works and sharing [...]”

“Open science infrastructures should be [...] essentially not-for-profit and long-term”

## EOSC SIRS report: Software Source Code and Open Science, 2020



- connect scholarly ecosystem via Software Heritage
- use open non profit infrastructures
- open source first: "all research software should be made available under an Open Source license by default"



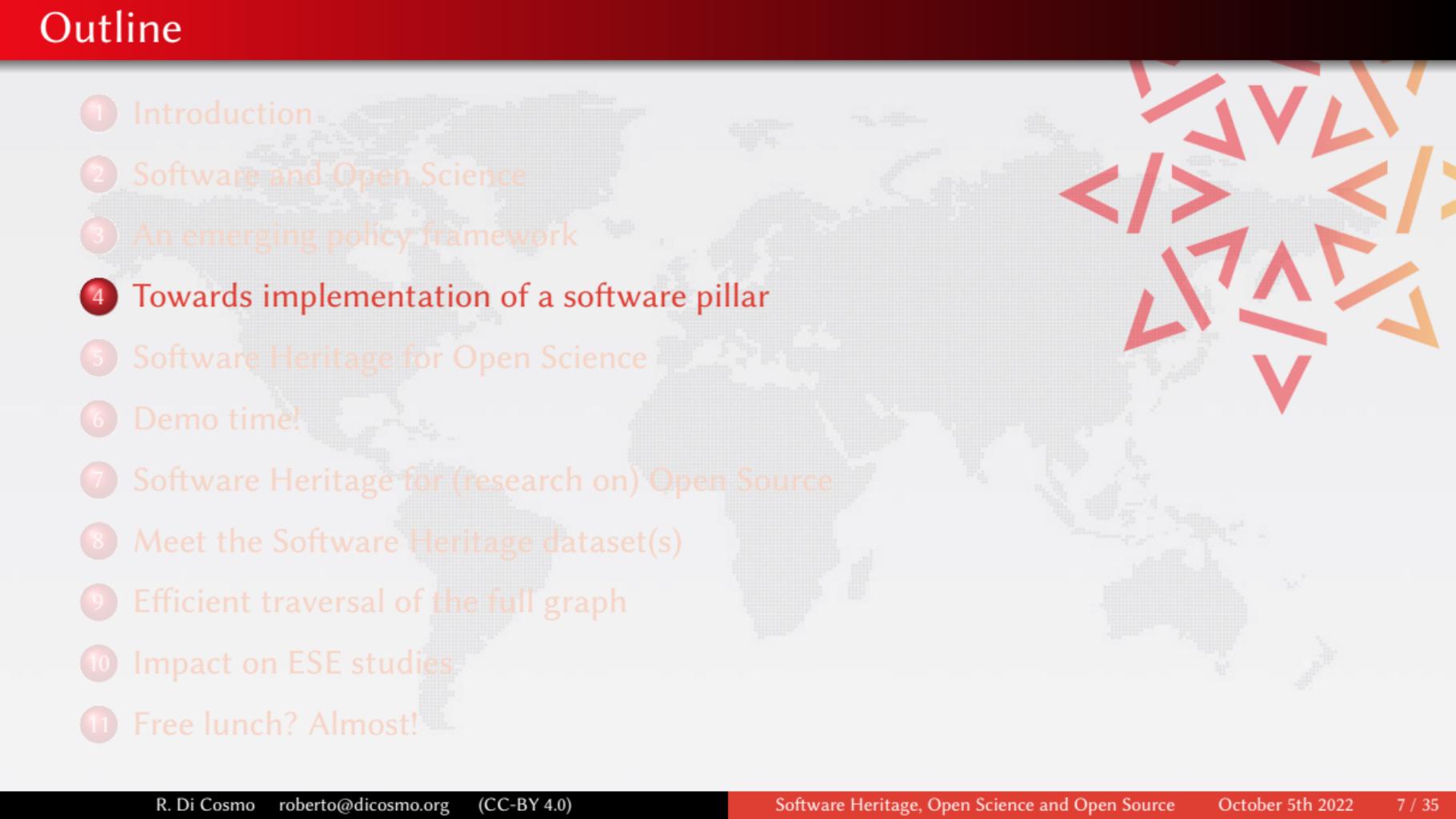
## 2nd National Plan for Open Science (6/7/2021)

### 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
  - 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](#)

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

# 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

# What is at stake

## ARDC

- **Archive** for retrieval (*reproducibility*)
- **Reference** for identification (*reproducibility*)
- **Describe** for discovery and reuse
- **Cite/Credit** for credit and evaluation

## Before ARDC

- **Development** practices and tools (VCS, build system, test suites, CI, ...)
- **Opening up** towards a community (documentation, organization, communication)

Need training, best practices

## Beyond ARDC

- **Policies** (dissemination, reuse, careers!)
- **Sustainability** (legal, economic etc.)
- Technology transfer
- Advanced technologies and tools (quality, traceability, etc.)

let's focus on infrastructures for ARDC

# 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)

## Big bad news keep coming in

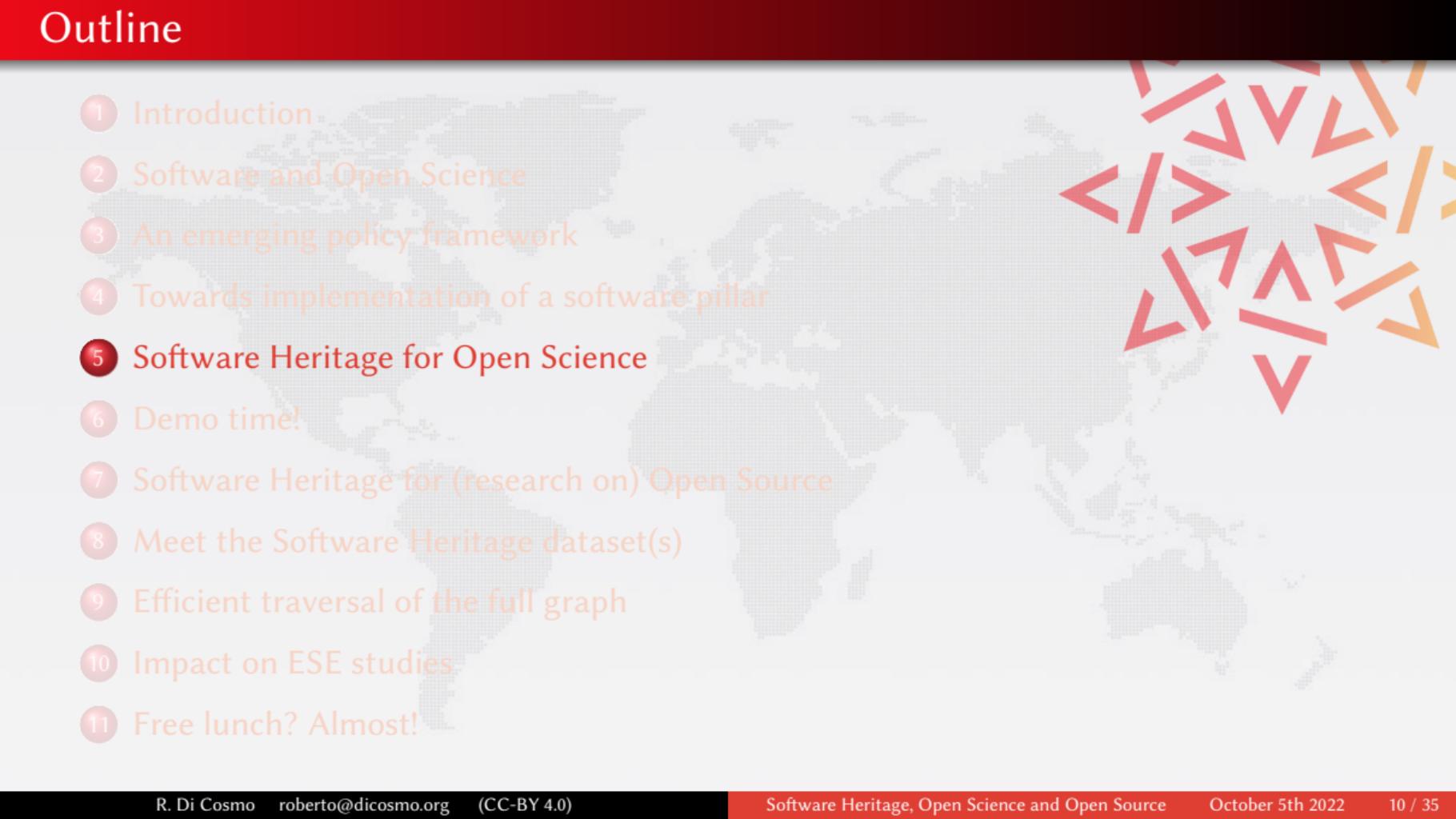
- summer 2019: BitBucket announces Mercurial VCS sunset
- july 2020: BitBucket erases 250.000+ repositories (including research software)
- summer 2022: GitLab.com considers erasing **all** projects that are **inactive for a year**

## In Academia too!

- 2021: Inria's old gforge is unplugged... **breaks the Opam build chain** for OCaml

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

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!



# 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

media  
aging  
tear  
attack  
malicious  
obsolete  
dependencies

damage  
disaster  
dangling  
deletion  
reference  
storage  
weird  
corruption  
encryption  
format

**preserve** all software  
source code

## Research infrastructure



enable analysis of all  
software source code

One infrastructure  
open and shared

Cultural Heritage      Industry      Research      Public Administration



## Software Heritage



Largest archive

## Technology

- transparency and FOSS
- replicas all the way down

## Content (billions!)

- intrinsic identifiers
- facts and provenance

## Organization

- non-profit
- multi-stakeholder

## Sharing the vision



United Nations  
Educational, Scientific and  
Cultural Organization



And many more ...

[www.softwareheritage.org/support/testimonials](http://www.softwareheritage.org/support/testimonials)

## Donors, members, sponsors



Diamond sponsor



Platinum sponsors



Gold sponsors

openinventionnetwork



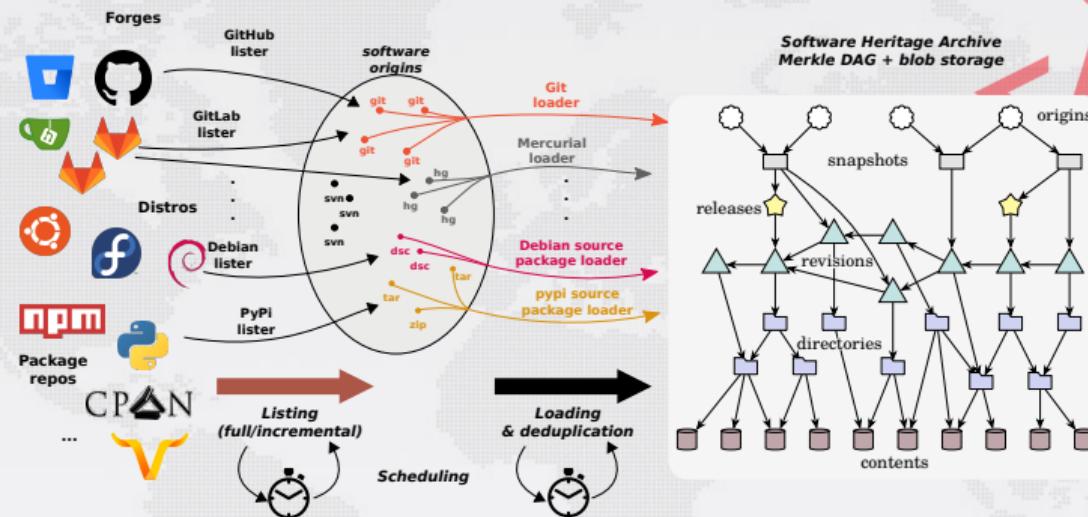
Silver sponsors



Bronze sponsors



# A peek under the hood: a universal archive



*Global development history permanently archived in a uniform data model*

- over **12 billion** unique source files from over **180 million** software projects
- ~1PB (uncompressed) blobs, ~25 B nodes, ~350 B edges

# A peek under the hood: listers and loaders

## Supported listers ([index](#))

Software Heritage - User Documentation

» Software Heritage listers

[View page source](#)

### Software Heritage listers

A **lister** is a software component used for the discovering of software origins to load into the Software Heritage archive.

This page references all available listers and links to their high-level documentation.

Lister name	Related links	Current status	Related grants
 Arch lister	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 AUR lister	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 Bitbucket lister	<ul style="list-style-type: none"><li>Source code</li><li>Developer doc</li><li>Development</li></ul>	in production	
 Bower lister	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	NLNet Foundation (awarded to Octobus)

CONTENTS:

- Frequently Asked Questions
- Software Heritage listers
  - Arch lister
  - AUR lister
  - Bitbucket lister
  - Bower lister
  - Cgit lister
  - CPAN lister
  - CRAN lister
  - Crates lister
  - Debian lister
  - Gitea lister
  - Github lister
  - GitLab lister

Search docs

## Supported loaders ([index](#))

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Search docs

[View page source](#)

### Software Heritage loaders

A **loader** is a software component used to ingest content into the Software Heritage archive.

This page references all available loaders and links to their high-level documentation.

Loader name	Related links	Current status	Related grants
 Arch loader	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 Archive loader	<ul style="list-style-type: none"><li>Source code</li><li>Developer doc</li></ul>	in production	
 AUR loader	<ul style="list-style-type: none"><li>Source code</li><li>Development</li></ul>	in development	Alfred P. Sloan Foundation (awarded to Hashbang)
 Bazaar loader	<ul style="list-style-type: none"><li>Source code</li><li>Developer doc</li><li>Development</li></ul>	in production	Alfred P. Sloan Foundation (awarded to Octobus)
 NPM loader	<ul style="list-style-type: none"><li>Source code</li></ul>		

Many contributed from external experts

thanks to support of Alfred P. Sloan and NLNet foundations

# Intrinsic Identifiers for software artefacts

## Software Heritage Identifiers (SWHID)

[link to full docs](#)

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

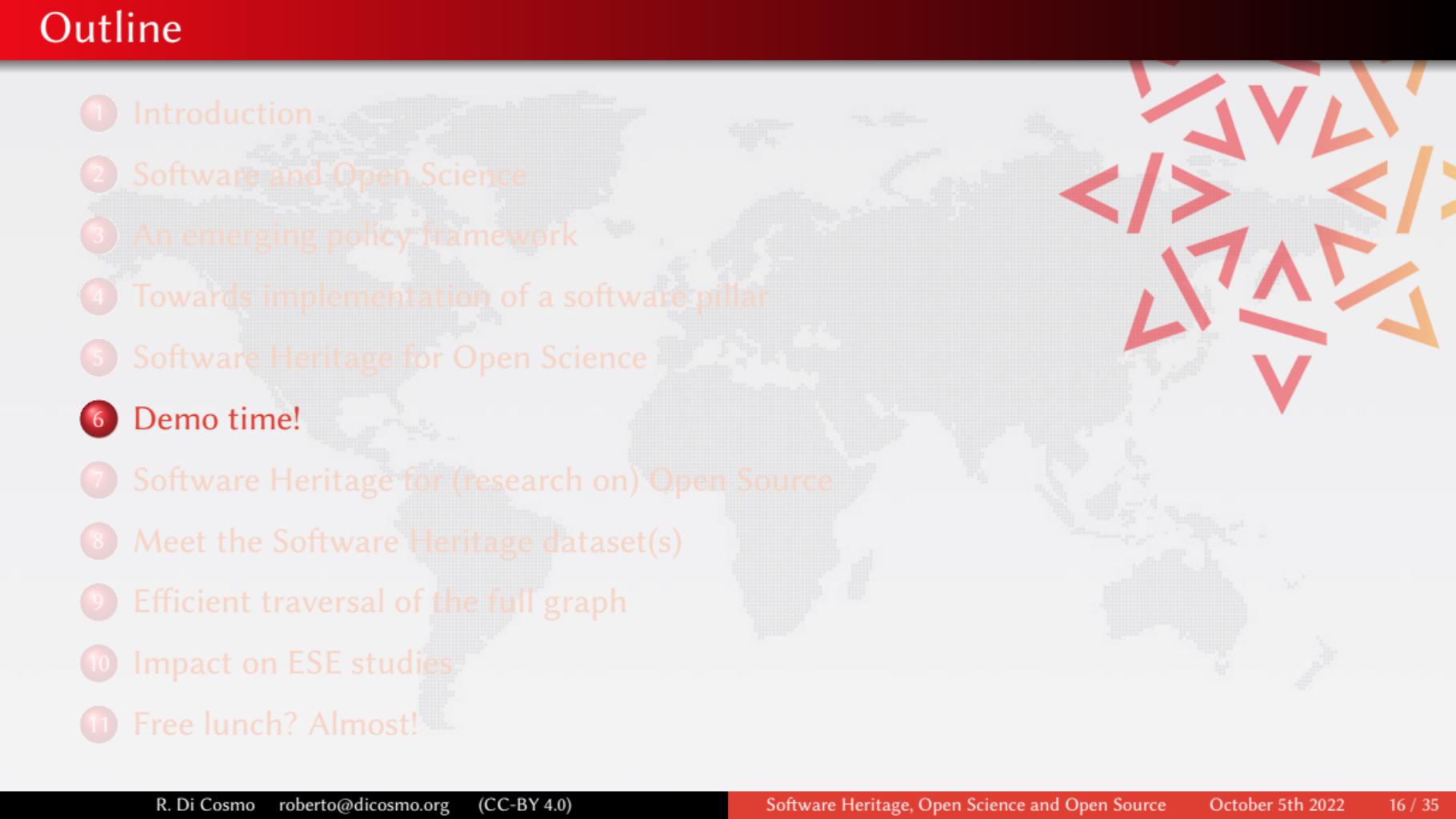


Emerging standard : Linux Foundation [SPDX 2.2](#); IANA registered; WikiData [P6138](#)

Full fledged *source code references* for reproducibility

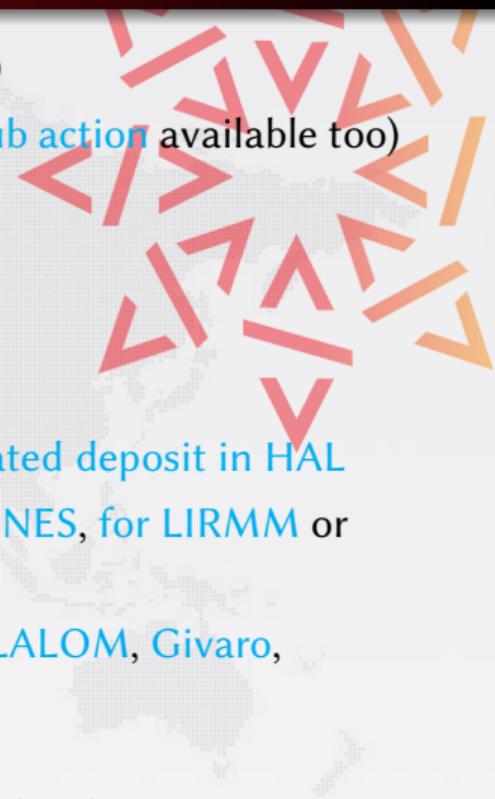
Examples: [Apollo 11 AGC excerpt](#), [Quake III rsqrt](#); Guidelines available, see [ICMS 2020](#)

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

# A walkthrough

- 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](#)



# Growing adoption of SWH in Academia (selection)

## HAL software curated deposit workflow

*Curated Archiving of Research Software Artifacts*

International Journal of Digital Curation, 2020

## IPOL (image processing)



- archive (deposit)
- reference
- BibLaTeX

## eLife (life sciences)



- archive (save code now)
- reference

## JTCAM (mechanics)

- instructions for authors
- biblatex-software in journal L<sup>A</sup>T<sub>E</sub>X class

## Policy: France

*National Plan for Open Science and Research Infrastructures*



## Policy: Europe



*EOSC SIRS report*

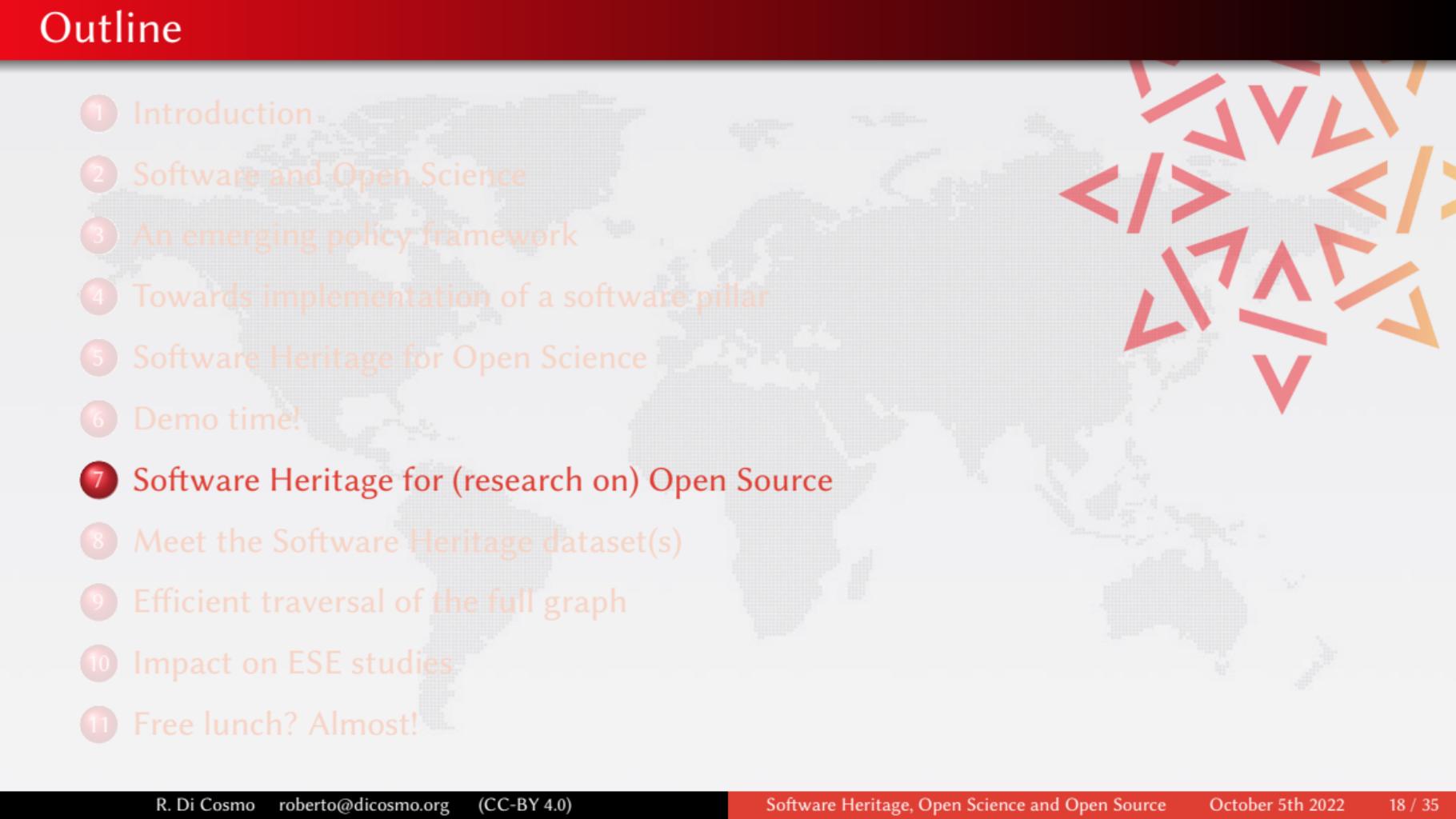
- SWHIDs
- archive

## Guidelines



- summary
- ICMS 2020

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

# Improving Security and Transparency for Open Source

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...

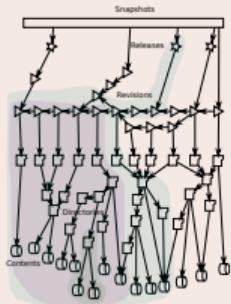


**Sec. 4. Enhancing Software Supply Chain Security**  
*ensuring and attesting, to the extent practicable, to the integrity and provenance of open source software*

May 2021 POTUS Executive Order

# A revolutionary infrastructure for industry

## The *graph* of Software Development

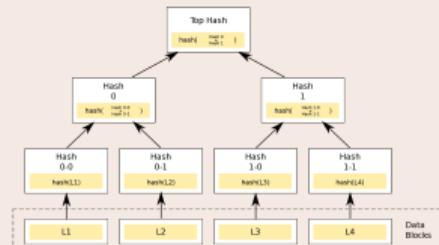


All of the software development in **a single graph!**

- **lookup** by content hash
- **wayback machine** for software development
  - <http://archive.softwareheritage.org/>
- ... and much more

## The *blockchain* of Software Development

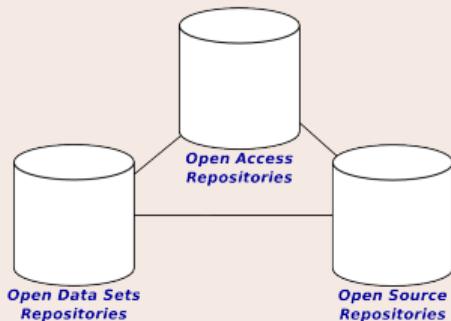
All of a software development... in a single **Merkle** graph!  
Widely used crypto (e.g., Git, blockchains, IPFS, ...)



- **built-in deduplication**
- **intrinsic, unforgeable identifiers** at all levels
- **simplifies traceability** (licensing, supply chain management)

# A revolutionary infrastructure for science

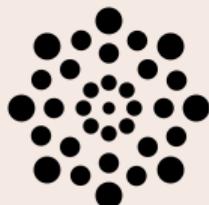
## A pillar of Open Science



The *reference archive* of Research Software for **Open Science**

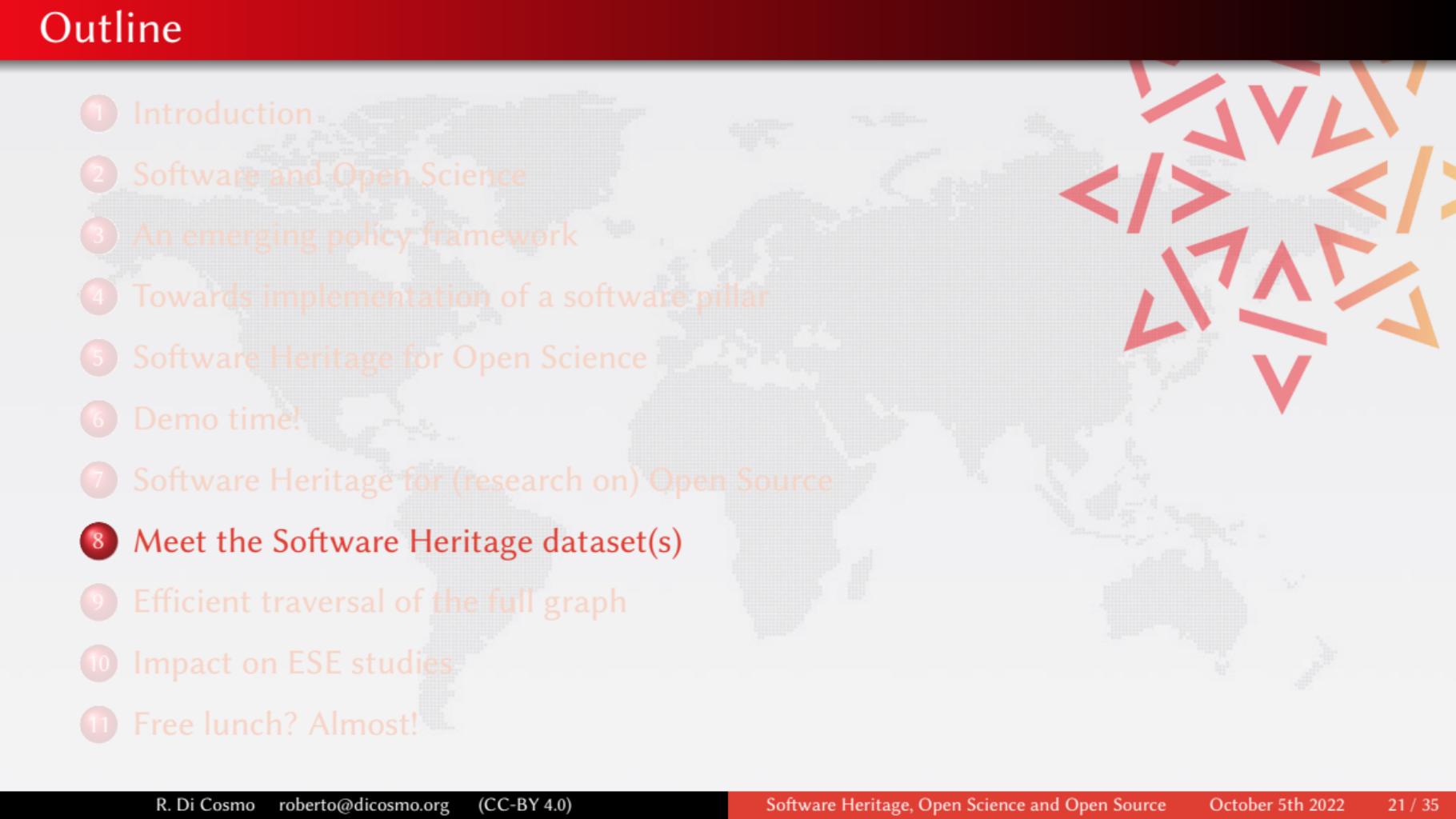
- curated deposit of research software
  - in collaboration with **HAL, CCSD** and **Inria IES**
  - now open *to all researchers!*
- **intrinsic** identifiers for **reproducibility**

## Reference platform for *Big Code*



- unique **observatory** of all software development
- **big data, machine learning** paradise: classification, trends, coding patterns, code completion...

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

# The full graph in the AWS Open Data collection

<https://registry.opendata.aws/software-heritage/>

Registry of Open Data on AWS



## Software Heritage Graph Dataset

[digital preservation](#) [free software](#) [open source software](#) [source code](#)

### Description

Software Heritage is the largest existing public archive of software source code and accompanying development history. The Software Heritage Graph Dataset is a fully deduplicated Merkle DAG representation of the Software Heritage archive. The dataset links together file content identifiers, source code directories, Version Control System (VCS) commits tracking evolution over time, up to the full states of VCS repositories as observed by Software Heritage during periodic crawls. The dataset's contents come from major development forges (including GitHub and GitLab), FOSS distributions (e.g., Debian), and language-specific package managers (e.g., PyPI). Crawling information is also included, providing timestamps about when and where all archived source code artifacts have been observed in the wild.

### Update Frequency

Data is updated yearly

### License

Creative Commons Attribution 4.0 International. By accessing the dataset, you agree with the Software Heritage [Ethical Charter](#) for using the archive data and the [terms of use for bulk access](#).

### Documentation

<https://docs.softwareheritage.org-devel/swh-dataset/graph/athena.html>

### Managed By

Software Heritage

See all datasets managed by [Software Heritage](#).

### Resources on AWS

#### Description

Software Heritage Graph Dataset

#### Resource type

S3 Bucket

#### Amazon Resource Name (ARN)

`arn:aws:s3:::softwareheritage`

#### AWS Region

`us-east-1`

#### AWS CLI Access (No AWS account required)

`aws s3 ls --no-sign-request s3://softwareheritage/`

#### Description

S3 Inventory files

#### Resource type

S3 Bucket

#### Amazon Resource Name (ARN)

`arn:aws:s3:::softwareheritage-inventory`

#### AWS Region

`us-east-1`

#### AWS CLI Access (No AWS account required)

`aws s3 ls --no-sign-request s3://softwareheritage-`

# A peek at the dataset

Accessing graph leaves (a.k.a. contents)

```
$ aws s3 ls --no-sign-request s3://softwareheritage/  
    PRE content/  
    PRE graph/
```

File contents can be accessed using their SHA1 checksum

```
$ aws s3 cp --no-sign-request \  
  s3://softwareheritage/content/8624bcdae55baeef00cd11d5dfcfa60f68710a02 .
```

Notice that file contents are compressed:

```
$ zcat 8624bcdae55baeef00cd11d5dfcfa60f68710a02 | head  
GNU GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007
```

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

# A peek at the dataset, cont'd

## Annual dumps of (inner nodes of) the full graph

```
$ aws s3 ls --no-sign-request s3://softwareheritage/graph/
```

2018-09-25/

2019-01-28-popular-3k-python/

2019-01-28-popular-4k/

2020-05-20/

2020-12-15/

2021-03-23-cpython-3-5/

2021-03-23-popular-3k-python/

2021-03-23/

2022-04-25/

## How to use

- [online full documentation](#)
- [Antoine Pietri's PhD Thesis](#)

## How to cite

Antoine Pietri, Diomidis Spinellis, Stefano Zacchiroli. *The Software Heritage Graph Dataset: Public software development under one roof*. MSR 2019. ([bibtex](#))

# Example: most popular commit verbs (stemmed)



## Query using Amazon Athena

```
SELECT COUNT(*) AS C, word FROM (
    SELECT word_stem(lower(split_part(
        trim(from_utf8(message)), ' ', 1)))
    AS word FROM revision
    WHERE length(message) < 1000000)
WHERE word != ''
GROUP BY word
ORDER BY C
DESC LIMIT 20;
```

*Total cost: approximately .5 euros*

## Results

Completed

Time in queue: 272 ms

Run time: 33.545 sec

Data scanned: 94.51 GB

Results (20)

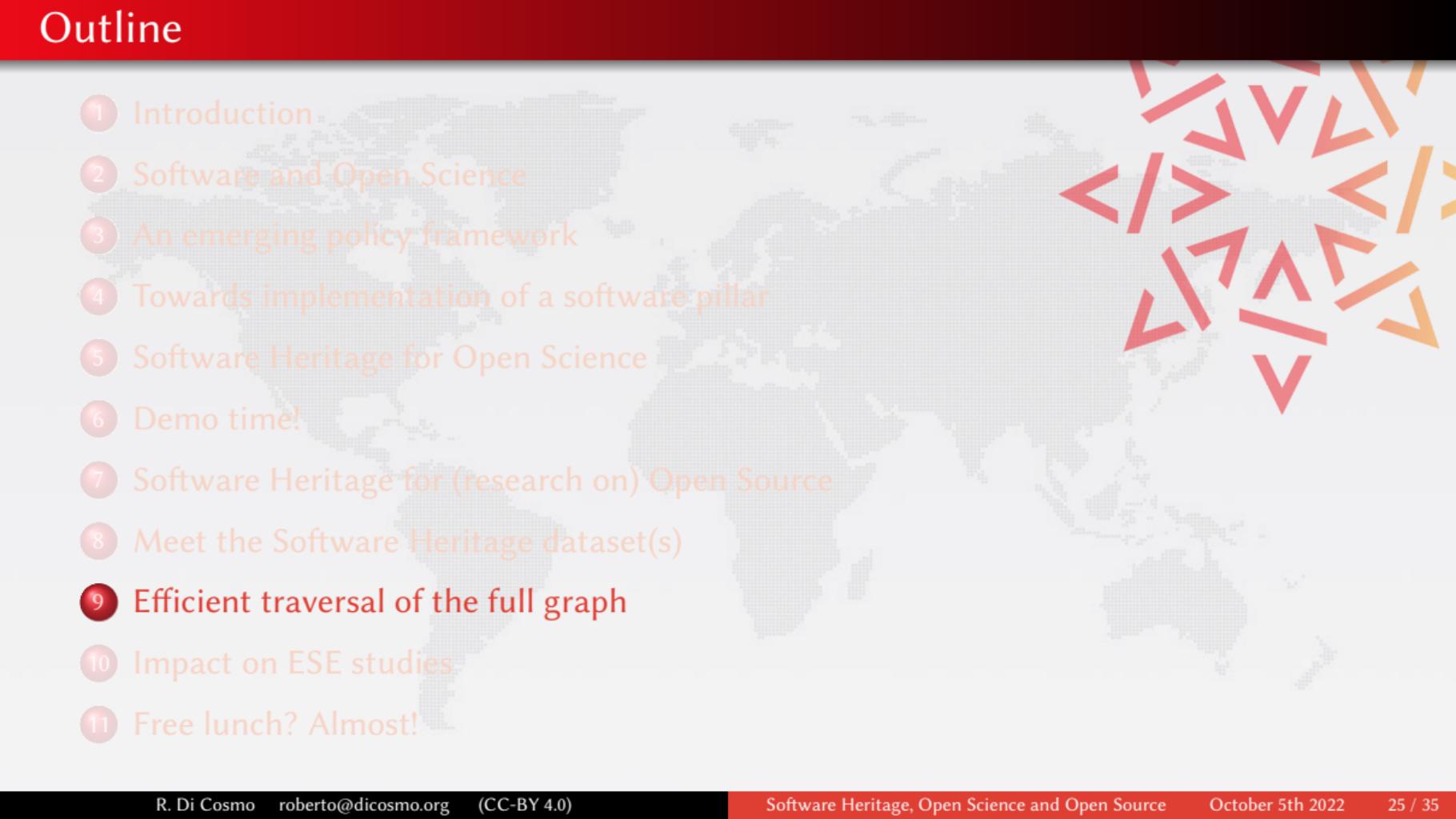
Copy

Download results

< 1 >

#	c	word
1	271573294	updat
2	163328012	merg
3	140044381	add
4	105800317	fix
5	103646653	ad
6	52891401	bump
7	50067041	initi
8	45609622	creat
9	42633225	remov
10	32230842	chang
11	23110410	delet
12	20734745	new
13	16644508	commit
14	15651821	test

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

## State-of-the-art graph compression from social networks

 Paolo Boldi, Antoine Pietri, Sebastiano Vigna, Stefano Zacchiroli

Ultra-Large-Scale Repository Analysis via Graph Compression

SANER 2020, 27th Intl. Conf. on Software Analysis, Evolution and Reengineering. IEEE

### Results

Full graph structure (25 B nodes, 350 B edges) in 200 GiB RAM

- traversal time is tens of ns per edge
- bidirectional traversals implemented
- **beware:** metadata access is still *off RAM*

### Java and gRPC APIs available

[docs.softwareheritage.org/devel/swh-graph/grpc-api.html](https://docs.softwareheritage.org-devel/swh-graph/grpc-api.html)

## Find all origins containing a given content

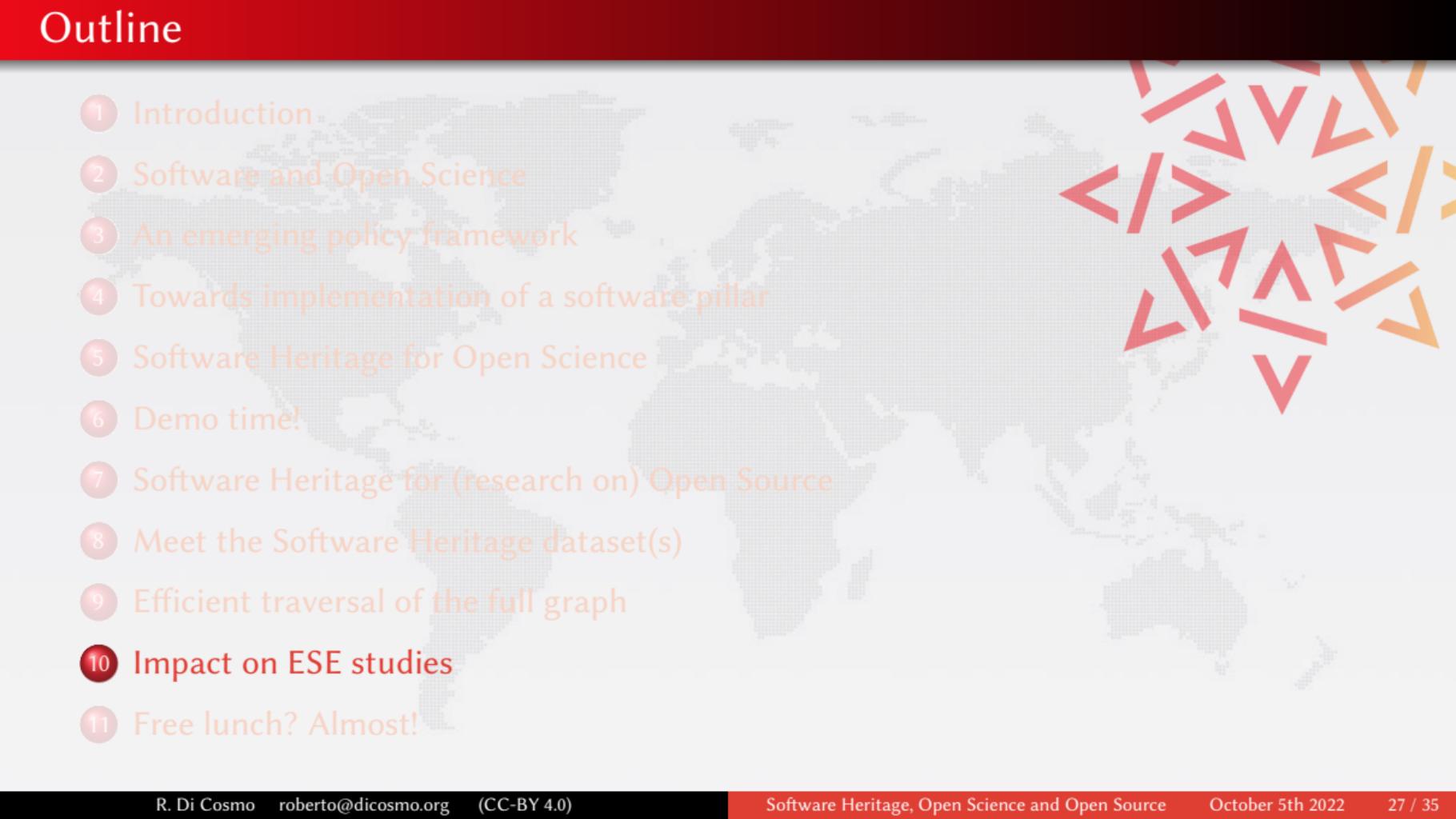
```
grpc_cli call localhost:50091 swh.graph.TraversalService.Traverse \"\nsrc: 'swh:1:cnt:8722d84d658e5e11519b807abb5c05bfbfc531f0', direction: BACKWARD,\nmask: {paths: ['swhid', 'ori.url']}, return_nodes: {types: 'ori'}\"
```

Gives a list of origins including "<https://github.com/rdicosmo/parmap>", encoded as  
"swh:1:ori:8903a90cff8f07159be7aed69f19d66d33db3f86" (**beware**: this is **not** a SWHID!)

## Shortest provenance path of a content in a given origin

```
grpc_cli call localhost:50091 swh.graph.TraversalService.FindPathBetween \"\nsrc: 'swh:1:ori:8903a90cff8f07159be7aed69f19d66d33db3f86', '\ndst: 'swh:1:cnt:8722d84d658e5e11519b807abb5c05bfbfc531f0', '\nmask: {paths: ['swhid']}\" | egrep 'swhid'\nconnecting to localhost:50091\n\nswhid: "swh:1:ori:8903a90cff8f07159be7aed69f19d66d33db3f86"\nswhid: "swh:1:snp:1527a93b039d70f6a781b05d76b77c6209912887"\nswhid: "swh:1:rev:82df563aecf86b9164eee7d10d40f2d8cbd1c78d"\nswhid: "swh:1:dir:484db39bb2825886191837bb0960b7450f9099bb"\nswhid: "swh:1:dir:4d15e44b378fe39dd23817abee756cd47ad14575"\nswhid: "swh:1:cnt:8722d84d658e5e11519b807abb5c05bfbfc531f0"\n\nRpc succeeded with OK status
```

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

# Selected research works using Software Heritage



**Thibault Allançon, Antoine Pietri, Stefano Zacchiroli**

The Software Heritage Filesystem (SwfFS): Integrating Source Code Archival with Development.

ICSE 2021: The 43rd International Conference on Software Engineering <https://arxiv.org/abs/2102.06390>



**Stefano Zacchiroli**

Gender Differences in Public Code Contributions: a 50-year Perspective

IEEE Softw. 38(2): 45-50 (2021)



**Antoine Pietri, Guillaume Rousseau, Stefano Zacchiroli**

Forking Without Clicking: on How to Identify Software Repository Forks

MSR 2020: 17th Intl. Conf. on Mining Software Repositories. IEEE



**Antoine Pietri, Guillaume Rousseau, Stefano Zacchiroli**

Determining the Intrinsic Structure of Public Software Development History

MSR 2020: 17th Intl. Conf. on Mining Software Repositories. IEEE



**Paolo Boldi, Antoine Pietri, Sebastiano Vigna, Stefano Zacchiroli**

Ultra-Large-Scale Repository Analysis via Graph Compression

SANER 2020, 27th Intl. Conf. on Software Analysis, Evolution and Reengineering. IEEE



**Roberto Di Cosmo, Guillaume Rousseau, Stefano Zacchiroli**

Software Provenance Tracking at the Scale of Public Source Code

Empirical Software Engineering 25(4): 2930-2959 (2020)

## Mining Android Applications on Software Heritage

*RQ: how to build a specific dataset for a given research question?*



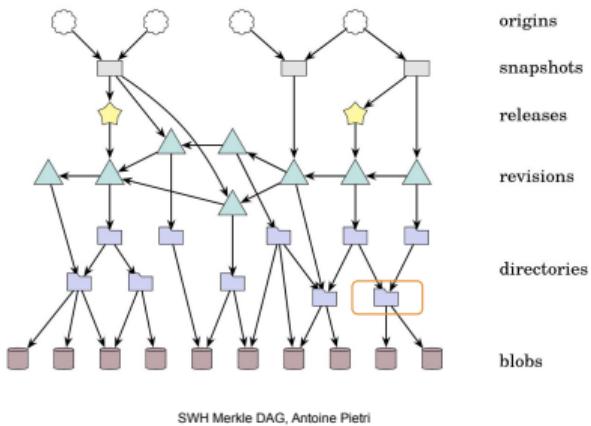
- Specific and limited API
- Hardly reproducible

- Generic, single, fine-grained and unlimited API
- Growing number of source codes
- Easy to update the dataset

(from the Inria/IRISA DiverSE team)

## Using the SWH merkle dag to identify android repositories

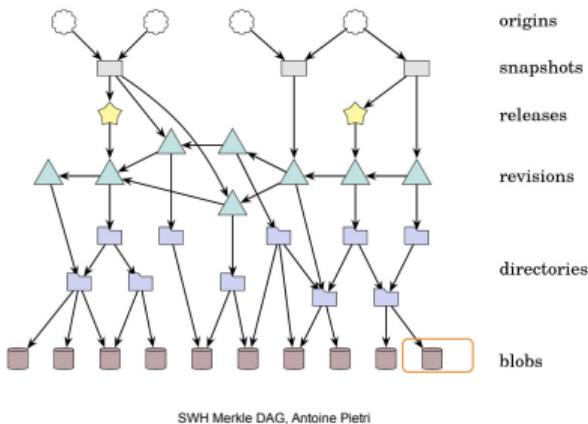
Identify android application repositories = Find the `AndroidManifest.xml` among the sources



- 1) Iterate over the graph nodes until you find a directory node containing a file named "AndroidManifest.xml".

## Using the SWH merkle dag to identify android repositories

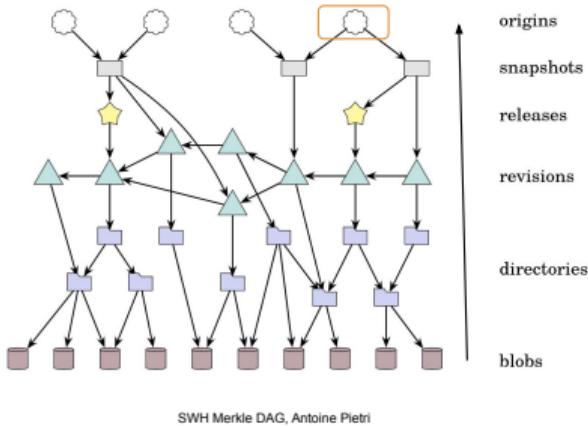
Identify android application repositories = Find the AndroidManifest.xml among the sources



- 2) Extract the SWH identifier of the blob corresponding to the `AndroidManifest.xml` and download the corresponding file through the SWH Web API

## Using the SWH merkle dag to identify android repositories

Identify android application repositories = Find the `AndroidManifest.xml` among the sources



3) Traverse the graph in backward direction to the origin node and get the repository url

Broad variety of sources in *one open dataset*

reduces usual GH bias

Reference simple *standard data format*

VCS and forge details are abstracted away

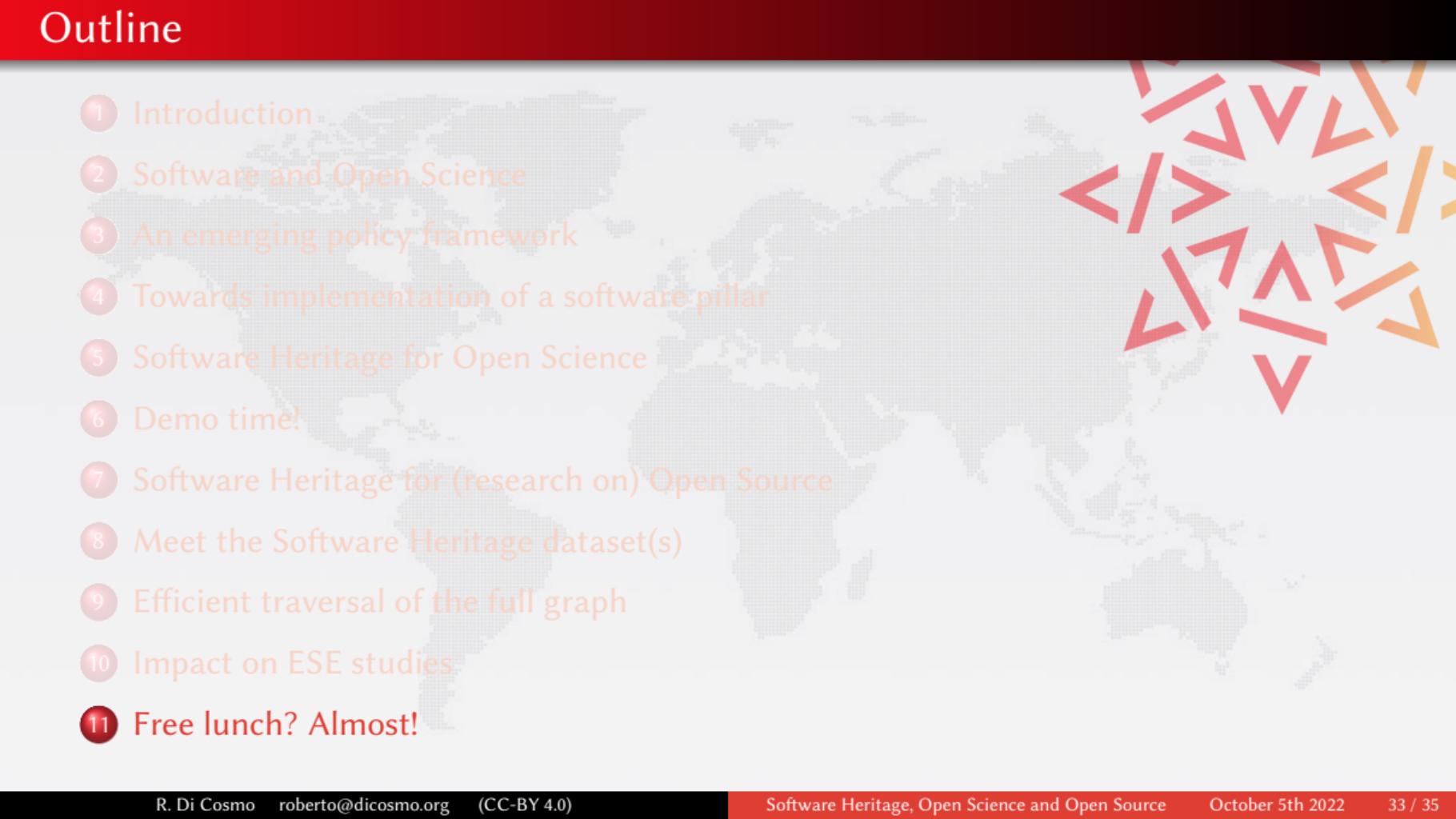
Simplifies reproducibility packages

no need to create a full copy, *just list the SWHIDs!*

Software Heritage does the heavy lifting for you

no need to scrape/download repositories all over again

# Outline

- 
- 1 Introduction
  - 2 Software and Open Science
  - 3 An emerging policy framework
  - 4 Towards implementation of a software pillar
  - 5 Software Heritage for Open Science
  - 6 Demo time!
  - 7 Software Heritage for (research on) Open Source
  - 8 Meet the Software Heritage dataset(s)
  - 9 Efficient traversal of the full graph
  - 10 Impact on ESE studies
  - 11 Free lunch? Almost!

# Call to action: best practices for ARDC are available... today!

## Archiving and referencing

For **all source code used in research (yes, even small scripts!)**

- archive and reference in Software Heritage (see [detailed HOWTO](#))

## Describing and Citing/Crediting

For **software one wants to put forward**, add these **extra steps**:

- add **codemeta.json** with description (see the [codemeta generator](#))
- (french partners) reference in the HAL portal (see [online HAL documentation](#))
- cite software using the [biblatex-software](#) package (in CTAN and TeXLive)

## We can (and must)

- train students and colleagues
- engage journals, conferences, learned societies

# 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!

help maintain and improve the infrastructure, adapt research tools to work with it, ...

Let's work together!

## 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 [\(10.1145/3183558\)](#)