

Software Heritage

The Great Library of (Python) Source Code

Nicolas Dandrimont, Stefano Zacchioli

Software Heritage – {olasd,zack}@softwareheritage.org

6 Oct 2018

PyConFr - Lille, France



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

- 
- 1 The Software Commons
 - 2 Software Heritage
 - 3 The Great Library of Python source code
 - 4 Getting involved

(Free) Software is everywhere



Software source code is *special*

Harold Abelson, Structure and Interpretation of Computer Programs

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

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;
}
```

Net. queue in Linux (excerpt)

```
/*
 * SFB uses two B[1][n] : L x N arrays of bins (L levels, N bins per level)
 * This implementation uses L = 8 and N = 16
 * This permits us to split one 32bit hash (provided per packet by rxhash or
 * external classifier) into 8 subhashes of 4 bits.
 */
#define SFB_BUCKET_SHIFT 4
#define SFB_NUMBUCKETS (1 << SFB_BUCKET_SHIFT) /* N bins per Level */
#define SFB_BUCKET_MASK (SFB_NUMBUCKETS - 1)
#define SFB_LEVELS (32 / SFB_BUCKET_SHIFT) /* L */

/* SFB algo uses a virtual queue, named "bin" */
struct sfb_bucket {
    u16 qlen; /* length of virtual queue */
    u16 p_mark; /* marking probability */
};
```

Len Shustek, Computer History Museum

“Source code provides a view into the mind of the designer.”

Definition (Commons)

The **commons** is the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable earth. These resources are held in common, not owned privately. <https://en.wikipedia.org/wiki/Commons>

Definition (Software Commons)

The **software commons** consists of all computer software which is available at little or no cost and which can be altered and reused with few restrictions. Thus *all open source software and all free software are part of the [software] commons.* [...]

https://en.wikipedia.org/wiki/Software_Commons

Definition (Commons)

The **commons** is the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable earth. These resources are held in common, not owned privately. <https://en.wikipedia.org/wiki/Commons>

Definition (Software Commons)

The **software commons** consists of all computer software which is available at little or no cost and which can be altered and reused with few restrictions. Thus *all open source software and all free software are part of the [software] commons.* [...]

https://en.wikipedia.org/wiki/Software_Commons

Source code is *a precious part of our commons*

are we taking care of it?



Fashion victims

- many disparate development platforms
- a myriad places where distribution may happen
- projects tend to migrate from one place to another over time

Software is spread all around



Fashion victims

- many disparate development platforms
- a myriad places where distribution may happen
- projects tend to migrate from one place to another over time

Where is the place ...

where we can find, track and search *all* source code?



A word cloud of terms related to software fragility and digital information loss. The words are arranged in a cluster, with 'damage' and 'disaster' being the largest. Other prominent words include 'malicious', 'deletion', 'obsolete', 'attack', 'format', 'dependencies', 'aging', 'tear', 'dangling', 'wear', 'corruption', 'encryption', 'reference', and 'storage'. The background features a faint world map and a decorative pattern of colorful triangles on the right side.

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code)
- for obsolete code: physical media decay (data rot)



A word cloud of terms related to software fragility, including: damage, disaster, malicious, deletion, obsolete, attack, dependencies, aging, media, tear, dangling, wear, corruption, encryption, format, reference, and storage. The words are arranged in a circular pattern with varying colors and sizes.

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code)
- for obsolete code: physical media decay (data rot)

Where is the archive...

where we go if (a repository on) GitHub or GitLab.com goes away?

Software lacks its own research infrastructure



A wealth of software research on crucial issues...

- safety, security, test, verification, proof
- software engineering, software evolution
- big data, machine learning, empirical studies

Software lacks its own research infrastructure

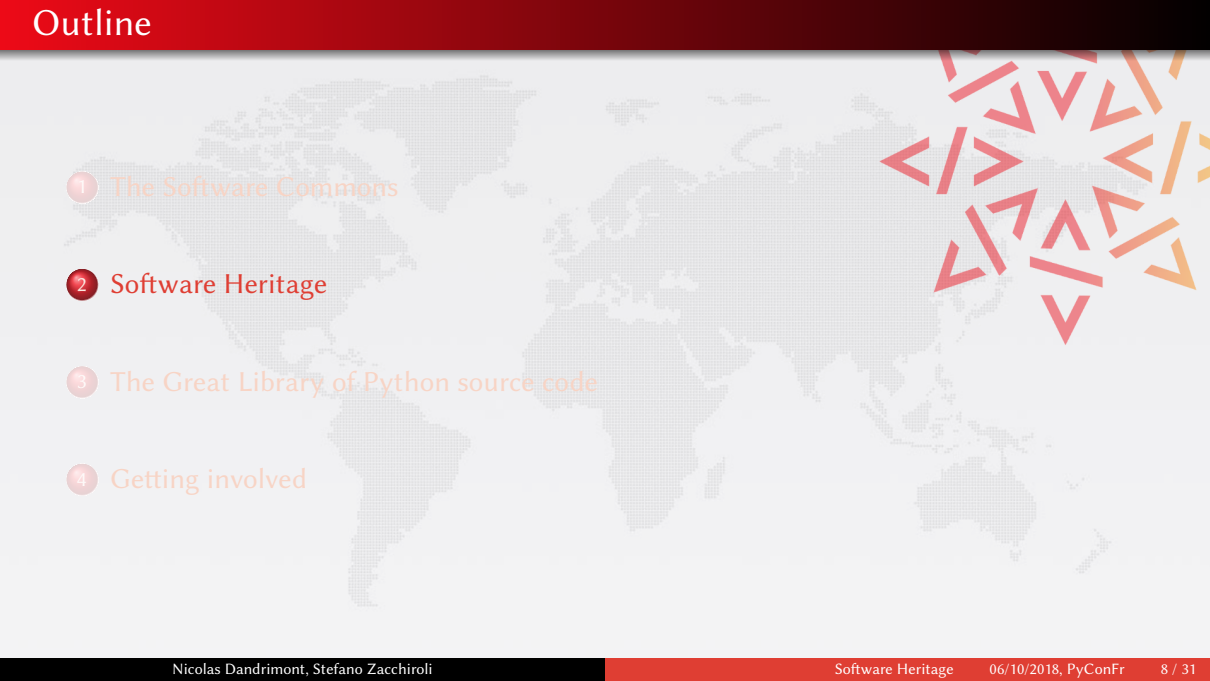


A wealth of software research on crucial issues...

- safety, security, test, verification, proof
- software engineering, software evolution
- big data, machine learning, empirical studies

If you study the stars, you go to Atacama...

... where is the *very large telescope* of source code?

- 
- 1 The Software Commons
 - 2 Software Heritage
 - 3 The Great Library of Python source code
 - 4 Getting involved



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Our mission

Collect, **preserve** and **share** the *source code* of *all the software* that is publicly available.

Past, present and future

Preserving the past, enhancing the present, preparing the future.

Cultural Heritage



Industry



Research



Education



Software Heritage

Cultural Heritage



Industry



Research



Education



Software Heritage

Open approach

- 100% Free Software
- transparency

In for the long haul

- replication
- non profit

Archiving goals

Targets: VCS repositories & source code releases (e.g., tarballs)

We DO archive

- file **content** (= blobs)
- **revisions** (= commits), with full metadata
- **releases** (= tags), ditto
- where (**origin**) & when (**visit**) we found any of the above

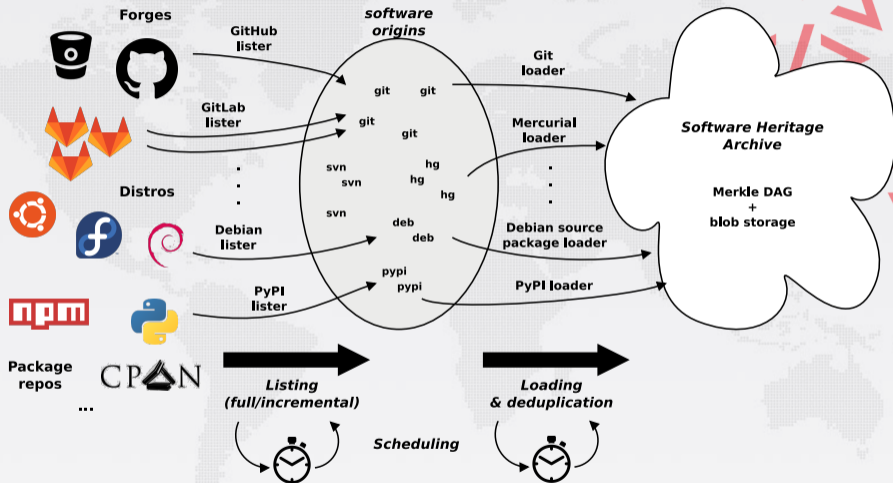
... in a VCS-/archive-agnostic **canonical data model**

We DON'T archive

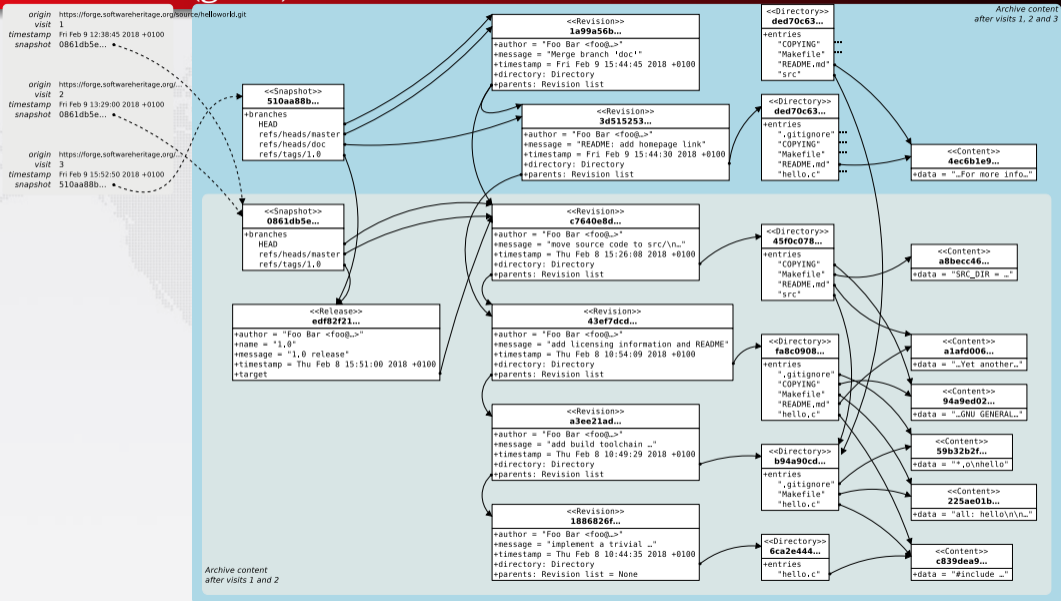
- homepages, wikis
- BTS/issues/code reviews/etc.
- mailing lists

Long term vision: play our part in a *"semantic wikipedia of software"*

Data flow



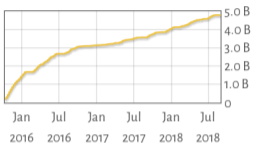
The archive: a (giant) Merkle DAG



Archive coverage

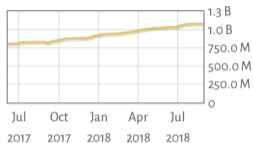
Source files

5,011,613,861



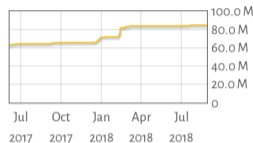
Commits

1,126,348,335



Projects

85,202,432



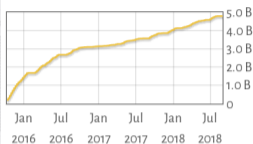
Current sources

- live: GitHub, Debian, GitLab.com, PyPI
- one-off: Gitorious, Google Code, GNU
- WIP: Bitbucket

Archive coverage

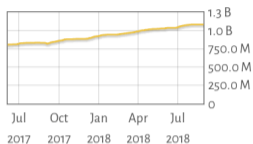
Source files

5,011,613,861



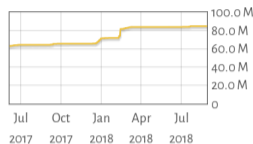
Commits

1,126,348,335



Projects

85,202,432



Current sources

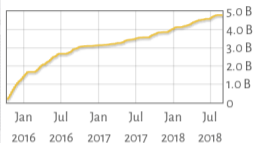
- live: GitHub, Debian, GitLab.com, PyPI
- one-off: Gitorious, Google Code, GNU
- WIP: Bitbucket

175 TB (compressed) blobs, 6 TB database (as a graph: 10 B nodes + 100 B edges)

Archive coverage

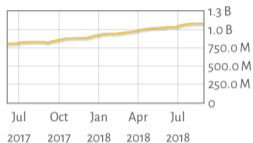
Source files

5,011,613,861



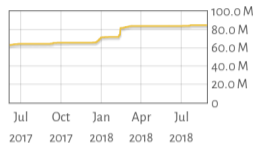
Commits

1,126,348,335



Projects

85,202,432



Current sources

- live: GitHub, Debian, GitLab.com, PyPI
- one-off: Gitorious, Google Code, GNU
- WIP: Bitbucket

175 TB (compressed) blobs, 6 TB database (as a graph: 10 B nodes + 100 B edges)

The *richest* public source code archive, ... and growing daily!

RESTful API to programmatically access the Software Heritage archive

<https://archive.softwareheritage.org/api/>

Features

- pointwise **browsing** of the archive
 - ... snapshots → revisions → directories → contents ...
- full access to the **metadata** of archived objects
- **crawling** information
 - *when have you last visited this Git repository I care about?*
 - *where were its branches/tags pointing to at the time?*

Endpoint index

<https://archive.softwareheritage.org/api/1/>

Vault service

- source code is thoroughly deduplicated within the Software Heritage archive
- bulk download of large artefacts (e.g., a Linux kernel release) requires collecting millions of objects
- the **Software Heritage Vault** cooks and caches source code bundles for bulk download needs

Tech bits

- **RESTful API** to request downloads, notifications, and monitoring
- `docs.softwareheritage.org/devel/swh-vault`

Browser-based interface to browse the Software Heritage archive

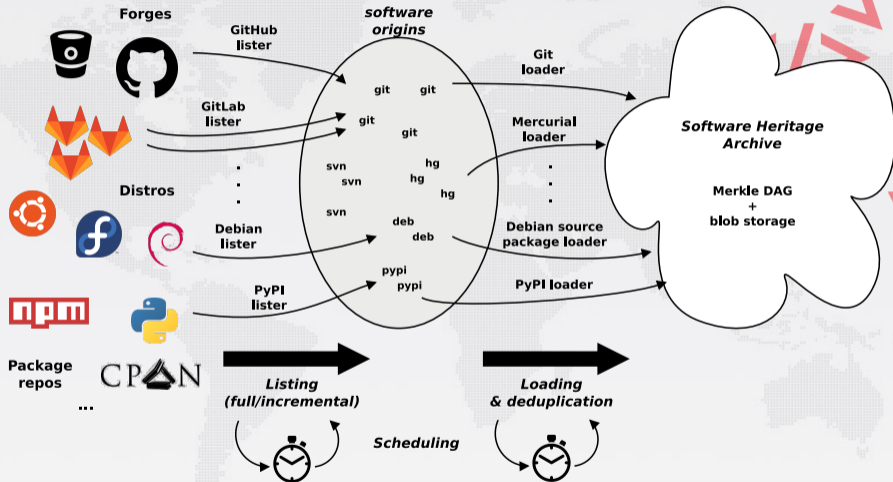
<https://archive.softwareheritage.org/browse/>

Features

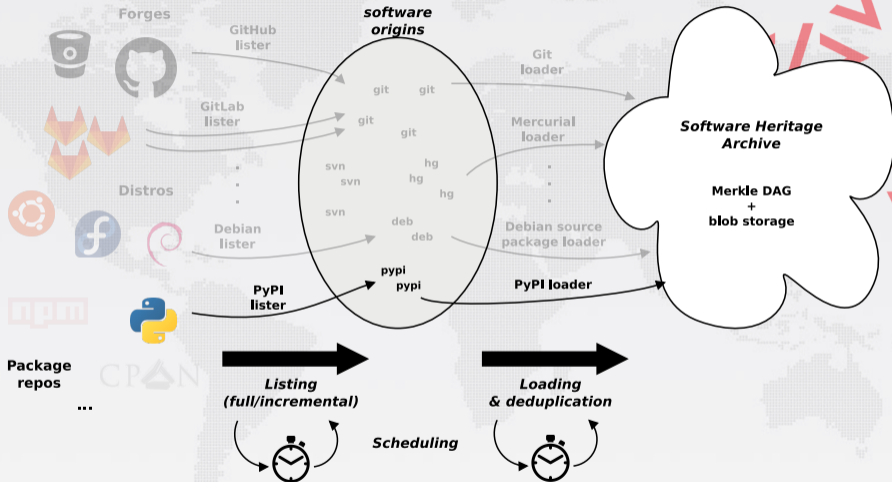
- all **REST API features**, but good looking :-)
 - browsing: snapshots → revisions → directories → contents ...
 - access to metadata and crawling information
- **origin search**, as full text indexing of origin URLs
- bulk **download**, via integration with the Vault

- 
- 1 The Software Commons
 - 2 Software Heritage
 - 3 The Great Library of Python source code
 - 4 Getting involved

Data flow redux



Our focus



Listing all Python modules (1/3)

<https://forge.softwareheritage.org/source/swh-lister/>

What does a Software Heritage lister do?

- crawls and parses upstream list of project APIs
- generates origins (records that the project has been detected) and loading tasks

Listing all Python modules (1/3)

<https://forge.softwareheritage.org/source/swh-lister/>

What does a Software Heritage lister do?

- crawls and parses upstream list of project APIs
- generates origins (records that the project has been detected) and loading tasks

Credits go to Avi Kelman for the lister scaffolding, and to Antoine Dumont for the PyPI implementation

Listing all Python modules (1/3)

<https://forge.softwareheritage.org/source/swh-lister/>

What does a Software Heritage lister do?

- crawls and parses upstream list of project APIs
- generates origins (records that the project has been detected) and loading tasks

Credits go to Avi Kelman for the lister scaffolding, and to Antoine Dumont for the PyPI implementation

A visit of the Cheese Shop

- A little bit more efficiently than John Cleese
- Uses <https://pypi.org/simple/> (according to the warehouse docs, the only "package listing" API that's not on the way to deprecation)

Listing all Python modules (2/3)

GET <https://pypi.org/simple/>

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Simple index</title>
5   </head>
6   <body>
7     <a href="/simple/0/">0</a>
8     <a href="/simple/0-0/">0-.....-0</a>
9     [...]
10    <a href="/simple/django/">Django</a>
11    [...]
12  </body>
13 </html>
```


Listing all Python modules (3/3)

```
1 # Origin specification
2 origin = {
3     'type': 'pypi',
4     'url': 'https://pypi.org/packages/Django/', # Canonical project URL
5 }
```

Listing all Python modules (3/3)

```
1 # Origin specification
2 origin = {
3     'type': 'pypi',
4     'url': 'https://pypi.org/packages/Django/', # Canonical project URL
5 }
6
7 # Scheduler task specification
8 update_task = {
9     'type': 'origin-update-pypi',
10    'policy': 'recurring',
11    'next_run': datetime.now(tz=timezone.utc),
12    'arguments': {
13        'args': [
14            'Django', # Project name
15            'https://pypi.org/packages/Django/', # Origin URL
16            'https://pypi.org/pypi/Django/json', # Metadata URL
17        ],
18        'kwargs': {},
19    },
20    'priority': None,
21 }
```

<https://forge.softwareheritage.org/source/swh-scheduler/>

What does the Software Heritage scheduler do?

- Record **recurrent** and **one-shot** jobs in a database
- Schedules runs of these jobs, records their results
- Manages retries for transient job failures (remote service unavailable, ...)
- Manages adaptive intervals for recurrent jobs

Task scheduling (2/2)

Builds upon trusted Python tools

- Celery is used as a task queuing middleware, and for its worker management framework
- Workers send task results through the Celery events mechanism

Builds upon trusted Python tools

- Celery is used as a task queuing middleware, and for its worker management framework
- Workers send task results through the Celery events mechanism

And makes them more useful to us

- The database is the single source of truth
- `swh.scheduler.celery_backend.runner` pulls tasks from the database into Celery, limiting the RabbitMQ queue depth (allows task prioritization)
- `swh.scheduler.celery_backend.listener` fetches task results from Celery events and updates the database
- Archival of elapsed tasks/runs/logs in elasticsearch to keep the database snappy

What's a Python package anyway?

- Source distributions (`sdist`s, currently tarballs or zips)
- Binary distributions (`bdist`s, which are mostly wheels these days)

As we're interested in source code, Software Heritage looks at `sdist`s exclusively

What's a Python package anyway?

- Source distributions (`sdist`s, currently tarballs or zips)
- Binary distributions (`bdist`s, which are mostly wheels these days)

As we're interested in source code, Software Heritage looks at `sdist`s exclusively

- The current `sdist` format is unspecified: you probably get a tarball, which maybe contains a `setup.py` somewhere
- When building a `sdist`, `distutils` generates a machine-readable `PKG-INFO` file is generated and puts in the tarball

Loading Python packages (1/4)

What's a Python package anyway?

- Source distributions (`sdist`s, currently tarballs or zips)
- Binary distributions (`bdist`s, which are mostly wheels these days)

As we're interested in source code, Software Heritage looks at `sdist`s exclusively

- The current `sdist` format is unspecified: you probably get a tarball, which maybe contains a `setup.py` somewhere
- When building a `sdist`, `distutils` generates a machine-readable `PKG-INFO` file is generated and puts in the tarball

The long wait for PEP 517 ("A build-system independent format for source trees")

- One uniform transport format: a gzipped tarball with one toplevel directory
- Machine parsable data about the project by default (`pyproject.toml`)

Hopefully soon in your nearest Cheese Shop (go help the folks in PyPA!)

<https://forge.softwareheritage.org/source/swh-loader-pypi/>

Common loading process

Implemented in `swh.loader.core`

- Fetch metadata about current versions
- Compare to latest loaded versions
- Download and process versions we had never seen
- Load new data

<https://forge.softwareheritage.org/source/swh-loader-pypi/>

Common loading process

Implemented in `swh.loader.core`

- Fetch metadata about current versions
- Compare to latest loaded versions
- Download and process versions we had never seen
- Load new data

PyPI specifics

Implemented in `swh.loader.pypi`

- Comparison done using the `sdist` digests
- PKG-INFO metadata parsed and saved
- versions with multiple `sdists` imported separately

PyPI snapshots

```
1 pifpaf_snapshot = {
2     'id': b'\xc6_\xfe#\x94\xba\x81\xc3\x94\x9b\xeb[\x06\xf5JC\xf0\x19n\xa6',
3     'branches': {
4         b'releases/0.0.1': {
5         b'releases/0.0.2': {
6             ...
7         b'releases/2.1.2': {
8             'target': b'\x8a\xcd\xf3\xee\xe5\xe2\x81]\x08:5\xd9_\xd6\reff\xc9\xa3',
9             'target_type': 'revision',
10        },
11        b'releases/2.1.2.dev7': {
12            'target': b'hGh\x15h|\xf3\xd2v\xf8\xec-\xa7\xfeuB\xda3\x83x',
13            'target_type': 'revision',
14        },
15        b'HEAD': {
16            'target': b'releases/2.1.2',
17            'target_type': 'alias',
18        },
19    },
20 }
```

PyPI revisions

```
1 pifpaf_revision = {
2     'id': b'\x8a\xcd\x31\xee\xe0\xe2\x81]\x08:5\xd9_\xd6\xeff\xc9\xa3',
3     'author': {
4         'name': b'Julien Danjou',
5         ...
6     },
7     'date': {
8         'timestamp': {'seconds': 1538577319, 'microseconds': 0},
9     },
10    ...
11    'type': 'tar',
12    'directory': b'\xa4\xf2\xad\xb1\xef\r\xcf\x894::@=\xf9R\x86=\x19"\',
13    'message': b'2.1.2',
```

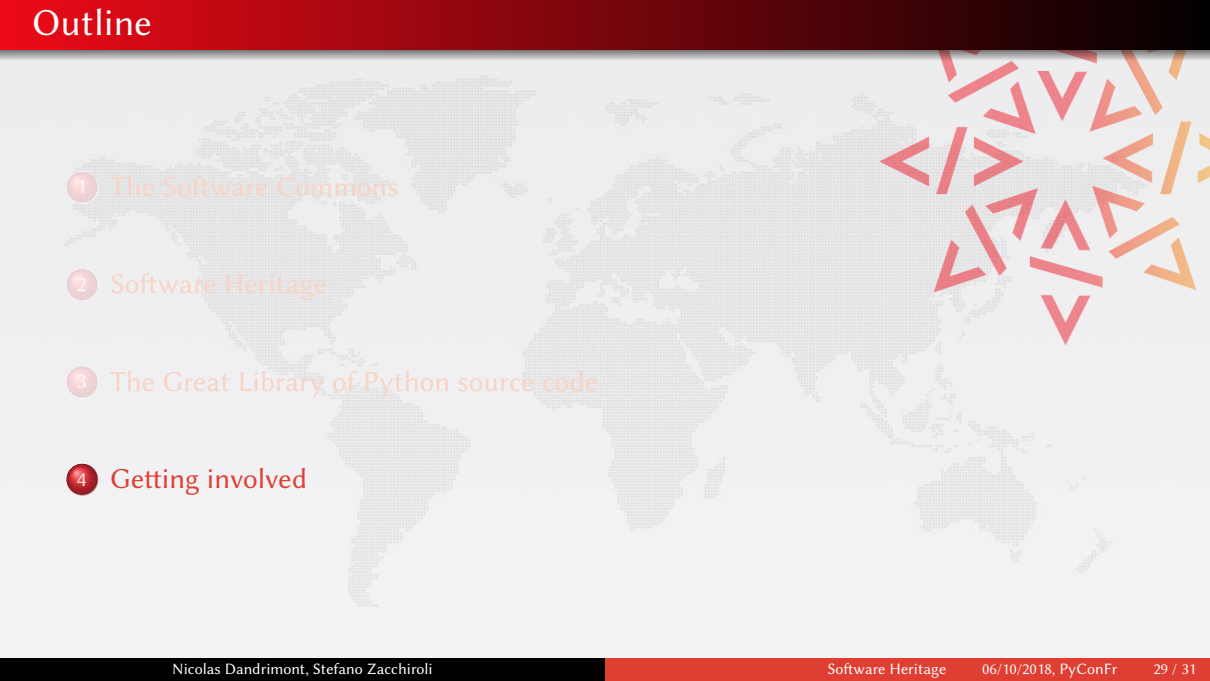
Loading Python packages (4/4)

PyPI revisions

```
1 pifpaf_revision = {
2     'id': b'\x8a\xcd\xf3l\xee\xe50\xe2\x81]\x08:5\xd9_\xd6\reff\xc9\xa3',
3     'author': {
4         'name': b'Julien Danjou',
5         ...
6     },
7     'date': {
8         'timestamp': {'seconds': 1538577319, 'microseconds': 0},
9     },
10    ...
11    'type': 'tar',
12    'directory': b'\xa4\xf2\xad\xb1\xef\r\xcf\x894::@=\xf9R\x86=\x19"\',
13    'message': b'2.1.2',
14
15    'metadata': {
16        'project': { # Metadata parsed from PKG-INFO
17            'name': 'pifpaf',
18            'author': 'Julien Danjou',
19            'license': None,
20            'summary': 'Suite of tools and fixtures to manage daemons for testing',
21            'version': '2.1.2',
22            ...
23        }
24    }
```

```
1     'classifiers': [  
2         'Intended Audience :: Information Technology',  
3         ...  
4     ],  
5     ...  
6 },
```

```
1     'classifiers': [  
2         'Intended Audience :: Information Technology',  
3         ...  
4     ],  
5     ...  
6 },  
  
1 'original_artifact': { # The original tarball we downloaded  
2     'url': 'https://files.pythonhosted.org/packages/cc/ce/2599[...]',  
3     'date': '2018-10-03T14:35:19',  
4     'sha1': '00c4efc47580b5c4ad1dcdb5118159f9b057b0fd',  
5     'size': 192940,  
6     'sha256': 'a6eef2ae56ac90d02df5f45885973e108c960a2ea113cc76[...]',  
7     'filename': 'pifpaf-2.1.2.tar.gz',  
8     'sha1_git': '8ce7e3ddda336dd9edff26ae8efaf4b81439c42c',  
9     'blake2s256': 'c4f7fcd4324715f4bfb54f8eeffb10fde803efb7a02e2[...]',  
10    'archive_type': 'tar',  
11 },  
12 },  
13 'synthetic': True,  
14 'parents': [],  
15 }
```

- 
- 1 The Software Commons
 - 2 Software Heritage
 - 3 The Great Library of Python source code
 - 4 Getting involved

Features...

- (done) **lookup** by content hash
- (done) **browsing**: "wayback machine" for source code (API + UI)
- (early access) **deposit** of source code bundles directly to the archive
- (early access) **save code now**, on-demand archive
- (done) **download**: `wget / git clone` from the archive
- (todo) **provenance** lookup for all archived content
- (todo) **full-text search** on all archived source code files

Features...

- (done) **lookup** by content hash
- (done) **browsing**: "wayback machine" for source code (API + UI)
- (early access) **deposit** of source code bundles directly to the archive
- (early access) **save code now**, on-demand archive
- (done) **download**: `wget / git clone` from the archive
- (todo) **provenance** lookup for all archived content
- (todo) **full-text search** on all archived source code files

... and much more than one could possibly imagine

all the world's software development history at hand's reach!

You can help!

Coding

- ★★ Web UI improvements
- ★★★ loaders for unsupported VCS/package formats
- ★★★ listers for unsupported forges/package managers

<https://forge.softwareheritage.org/>
<https://docs.softwareheritage.org/devel/>

You can help!

Coding

- ★★ Web UI improvements
- ★★★ loaders for unsupported VCS/package formats
- ★★★ listers for unsupported forges/package managers

<https://forge.softwareheritage.org/>
<https://docs.softwareheritage.org/devel/>

Community

- ★★★ spread the world, help us with sustainability
- ★★ document endangered source code

wiki.softwareheritage.org/Suggestion_box

You can help!

Coding

- ★★ Web UI improvements
- ★★★ loaders for unsupported VCS/package formats
- ★★★ listers for unsupported forges/package managers

<https://forge.softwareheritage.org/>
<https://docs.softwareheritage.org/devel/>

Community

- ★★★ spread the world, help us with sustainability
- ★★ document endangered source code

wiki.softwareheritage.org/Suggestion_box

Join us

- www.softwareheritage.org/jobs – **job openings**
- wiki.softwareheritage.org/Internship – **internships**

Software Heritage is

- a reference archive of **all Free Software** ever written
- an international, open, nonprofit, **mutualized infrastructure**
- **now accessible** to developers, users, vendors
- at the service of our community, **at the service of society**

Come in, we're open!

`www.softwareheritage.org` – general information
`wiki.softwareheritage.org` – internships, leads
`forge.softwareheritage.org` – our own code