



A NOVEL WAY TO EFFICIENTLY COMPLY WITH LICENSES

Alexios Zavras

Open Source Compliance
Intel Corp.

Software Licenses

FOSS licenses stipulate

- What you *may* do (permissions)
 - e.g., copy the code, modify it, re-distribute it
- What you *must* do (obligations)
 - e.g., use the same license, mention author's name

Source Code Distribution Obligation

Legal requirement: You have to deliver the source code along with the binary

- “For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable.”
 - GPLv2

Complete Corresponding Source (CCS)

Different terms used

- GPLv2: “complete corresponding machine-readable source code” / “accompany”
- GPLv3: “Corresponding Source” / “convey”
- MPLv2: “Source Code Form” / “made available”
- EPLv2: “Source Code” / “made available”

The Problem

In an ideal world

- Fool-proof processes in place
 - collect
 - package
 - provision
- Set it up once
 - always working
 - everyone uses this

In the real world

- People change roles or leave
- Re-organizations happen
- Things get forgotten

Use Cases

Combinations of own and external software:

1. Our delivery contains our own FOSS `sw.tar.gz`
2. Our delivery contains `gcc-8.2`
3. Our delivery contains `gcc` snapshot of revision 267928
4. Our delivery contains `gcc-8.2` patched with `patches.tar.gz`

Note: not only because of license obligation!

Functional Requirements

We need to be able to:

1. provide our own software package
2. refer to a “well-known” FOSS component
 - with release version or unique revision
3. combine the two
 - well-known component with own patches

Great Idea

Can we *outsource*
the fulfilment of our obligations?

Is It Legal?

GPL FAQ: Can I put the binaries on my Internet server and put the source on a different Internet site?

- [v3] Yes. Section 6(d) allows this. However, you must provide clear instructions people can follow to obtain the source, and you must take care to *make sure that the source remains available* for as long as you distribute the object code.
- [v2] The GPL says you must offer access to copy the source code “from the same place”; that is, next to the binaries. However, if you *make arrangements with another site* to keep the necessary source code available, and put a link or cross-reference to the source code next to the binaries, we think that qualifies as “from the same place”.

Great Idea

Wouldn't it be great if *someone* could fulfill our requirements?



Software Heritage

Mission:

Collect, preserve, and share the source code of all publicly available software

Non-profit

Coverage

Crawling of forges



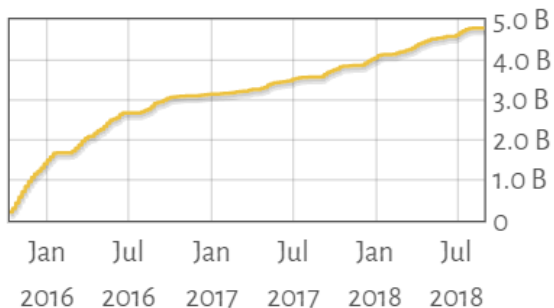
Size

The richest public source code archive

- 200 TB compressed blobs, 8 TB database (graph: 10B nodes, 100B edges)
 - Entirely de-duplicated

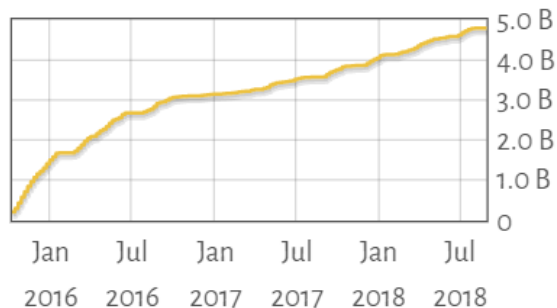
Source files

5,685,682,899



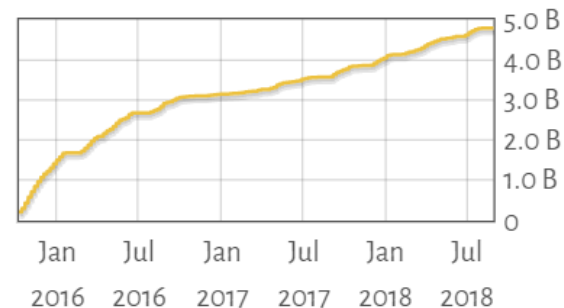
Commits

1,262,321,770



Projects

88,291,321



Archive Code

Besides the crawling of forges...

- “Add repository” to crawling
- “Take snapshot”
- “Direct deposit” (push)
 - SWORD 2.0 compliant
 - RESTful API

Once Archived... Identifiers

Anything on the archive

- Has a persistent identifier
 - e.g., `swh:1:cnt:0fffd12d85cdec70c88e852fc3f5ea9fd342213cd`
 - or `swh:1:dir:db990da9af15427455ce7836ce2b8a34b9bf67f5`
- Can be browsed online
- Can be downloaded

Conclusions

Obligations to deliver source code

- Legally compliant to outsource it
- Archive of everything available
- Allows adding own code
- Provides way to reference and retrieve

Problem solved!

Next Steps

From POC to Production service

- Gradual deployment
- Integration with software production and delivery
- Fine-tuning
 - check whether already available vs. upload everything
 - software optimization and customization

THANK YOU

Questions, comments, ... ?

