Complying with Open Source Licenses

Kate Stewart, Senior Director of Strategic Programs
Steve Winslow, Director of Strategic Programs
Three-way Handshake Needed

Management

Legal

Engineering

Triquetra Cross Image: By Madboy74 - Own work, CC BY-SA 4.0
Legal

Defining Open Source License Compliance
Open Source Licenses

Focusing on the Legal part of the partnership:

A legal mechanism intended to broadly enable rights for:

- **using** software
- **modifying** software
- **distributing** software

within a legal system that wasn’t designed with these goals in mind
Open Source Licenses

Copyright laws automatically give exclusive rights to authors of software.

They can grant licenses to allow others to exercise some of those rights.

Those licenses are often distinguished as “proprietary” vs. “open source”.
Open Source Licenses

**Proprietary** licenses:

- Do not grant the full set of rights associated with open source software
- Typically written for a single product or piece of software
- Often provide rights only for the “binary” version of software (not source code)
- Often treat software acquisition as a single event, from producer to consumer
  - Might be a lengthy click-through EULA
  - Might be heavily negotiated between attorneys (for B2B contracts)
  - High transaction costs
- Typically impose significant **restrictions** on use and redistribution
Open Source Licenses

**Open Source** licenses:

- Grant broad rights to use, modify, and distribute software - binary AND source
- Typically written for use with any piece of software
- Standardized (more or less) - small set of most commonly-used licenses
- Many will treat software acquisition as an ongoing series of events
  - Recognize that licensees can be not just consumers but also producers
  - Intended to result in small transaction costs
- Typically impose **responsibilities, not restrictions**, on redistribution or similar actions: “IF you redistribute, THEN you must do the following…”
Open Source License Types (broadly speaking)

Lesser obligations

Greater obligations
Open Source License Types (broadly speaking)

Lesser obligations

**Permissive**
(aka “Attribution”)

Main obligation:
If you redistribute the software, also provide its license and copyright notices.

Greater obligations
Open Source License Types (broadly speaking)

Lesser obligations

Permissive
(aka “Attribution”)

Main obligation:
If you redistribute the software, also provide its license and copyright notices.

Greater obligations

Copyleft
(aka “Reciprocal”)

Main obligation:
If you redistribute the software, also provide the same freedoms / rights to downstream recipients.
Open Source License Types (broadly speaking)

Lesser obligations

Permissive
Main obligation:
If you redistribute the software, also provide its license and copyright notices.

Weak Copyleft
Main obligation:
Similar to Strong Copyleft, with differences in the boundaries for the software to which the copyleft obligations apply.

Strong Copyleft
Main obligation:
If you redistribute the software, also provide the same freedoms / rights to downstream recipients.

Greater obligations
Open Source License Types (broadly speaking)

 Lesser obligations                           Greater obligations

Permissive   Weak Copyleft        Strong Copyleft

Common examples:
- BSD-2-Clause
- BSD-3-Clause
- MIT
- Apache-2.0

Common examples:
- GNU Lesser General Public License (LGPL)
- Mozilla Public License (MPL)
- Eclipse Public License (EPL)
- Common Development and Distribution License (CDDL)

Common examples:
- GNU General Public License (GPL)
- GNU Affero GPL (AGPL)
# Open Source License Types (broadly speaking)

<table>
<thead>
<tr>
<th>Lesser obligations</th>
<th>Greater obligations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Permissive</strong></td>
<td><strong>Copyleft</strong></td>
</tr>
<tr>
<td>Creative Commons Attribution 4.0 (CC-BY-4.0)</td>
<td>Creative Commons Attribution 4.0 ShareAlike (CC-BY-SA-4.0)</td>
</tr>
<tr>
<td>Community Data License Agreement – Permissive (CDLA-Permissive-1.0)</td>
<td>Community Data License Agreement – Sharing (CDLA-Sharing-1.0)</td>
</tr>
</tbody>
</table>

**Documentation and Creative Content**

**Data**
Open Source License Types (broadly speaking)

Lesser obligations  {distance} Greater obligations

Permissive

Copyleft

More considerations than just these, e.g.:

- Scope of licenses granted (copyright, patents, ...)
- Restrictions on prohibiting other rights (reverse engineering, ...)
- Other types of obligations (advertising clauses, ...)

But this spectrum reflects the major responsibilities people generally focus on for open source license compliance obligations.
Open Source License Compliance

Open Source license compliance is about fulfilling the responsibilities specified in the licenses, for the open source software that you’re using and/or distributing.

Compliance and legal considerations can also be broader than just satisfying license terms - e.g. export controls, patents, data privacy, …
Open Source License Compliance

But: Compliance is not just about minimizing or mitigating legal risks.

Also about being a participating member in communities that are building these technologies.

Respecting the efforts of those who are providing their work freely, for the world to benefit from.
Open Source License Compliance

Typical high-level process:

1. Identifying software and dependencies
2. Identifying licenses
3. Understanding contexts of use
4. Addressing any incompatibilities
5. Communicating license information
6. Providing source code, if required
Open Source License Compliance

Policies:

• What OSS licenses may we use in our products? in our tech stack?
• Under what circumstances will we open source our own code?

Processes:

• When and how often will we review licenses for open source dependencies in our products?
• What scanning and compliance tools will we use?
• Who will make decisions about what licenses are permissible in which circumstances?
• Where and how will we provide attribution notices and source code?
Open Source License Compliance

Compliance policies and procedures will mature over time. Every organization goes through a learning curve as they become more immersed in open source.

Continual improvement: get a little better every month; learn from experience of others

Effective compliance requires cross-functional partnership: In order for legal to make useful decisions and develop useful policies, must have buy-in and involvement from management and engineering teams.
Managing the risk of using open source software
Management: Accurate Inventory

This is not an insurmountable challenge

• Summaries are provided in a **Software Bill of Materials (SBOM)**
• Avoid making “perfect” the enemy of “better”

Tackling it benefits projects and benefits the whole ecosystem

• There are gaps in today’s tooling but there is also forward progress
Information relevant for Management

Summary of license and vulnerability findings, tailored for discussion with legal counsel and executives for risk management.

May include expert recommendations and dynamic generated information to assist with evaluating requests.
Open Source License Compliance

Typical high-level process:

1. Identifying software and dependencies
2. Identifying licenses
3. Understanding contexts of use
4. Addressing any incompatibilities
5. Communicating license information
6. Providing source code, if required
Accurate Identification of Software

• Necessary for Supply Chain Management (SCM):
  – Incoming and in active internal deployment
  – Outgoing and distribution requirements

• Necessary for Effective Software Vulnerability management

• Requires tracking relevant open source dependencies.
  – Different programming languages impose different standard ways of identifying and handling third party dependencies.
  – Adds complexity when thinking about what components are “in” the project code.

⇒ Prerequisites for Managing Risk and Compliance with Licenses
How? Emerging Best Practices

**OpenChain** (Process between organizations)

**SParts** (Distributed ledger for logging and tracking in supply chain)

**REUSE** (Artifact organization in projects)

**Core Infrastructure Initiative** (Security and Transparency)
The OpenChain Project documents the processes to build trust between members of a software supply chain using open source software.
SParts (Software Parts Ledger) uses blockchain technology to manage open source across the supply chain. Utilizes Hyperledger Sawtooth Platform and SPDX-based SBOM to conform to OpenChain best practices. See: https://github.com/Wind-River/sparts

Accepted into Hyperledger Labs, Mar. 2018: https://github.com/hyperledger-labs/hyperledger-labs.github.io/blob/master/labs/SParts.md
The **REUSE Initiative**, developed by FSFE, describes best practices for describing licensing information in open source software and making it suitable for automation.

See: [https://reuse.software/](https://reuse.software/)

Practices: [https://reuse.software/practices/2.0/](https://reuse.software/practices/2.0/)

Overview: [https://reuse.software/reuse/reuse-presentation.pdf](https://reuse.software/reuse/reuse-presentation.pdf)
Best Practices: In Projects

Core Infrastructure Initiative (CII) Best Practices –
To achieve Gold badge status, accurate copyright and licensing information is required.

See, e.g.:
Figuring out what open source software you have...
Information relevant to Engineering

List of packages being used

Include name, version number, checksums, download location, source location, license information, build / run dependencies, vulnerability identifiers, etc.

Tooling is needed to build up a product and track components.
Open Source License Compliance

Typical high-level process:

1. Identifying software and dependencies
2. Identifying licenses
3. Understanding contexts of use
4. Addressing any incompatibilities
5. Communicating license information
6. Providing source code, if required

GOAL: Automate as much of this as possible
What goes into product SBOM?

Companies combine OSS with other software

Software Bill of Materials (BOM)
Supply Chain Impact

Supplier 1

Supplier 2

Customers

The effort is repeated at each step in the supply chain
Identifying Relevant Software Information

- Software Bills of Materials (SBOMs)
  - Sources, binaries
- Company database of software in use
  - Facts tracked for ingress & created software
- External metadata repos?
- Upstream projects?
Identifying Software in Use: Scanning Tools

Various scanning tools and services exist, including open source and proprietary / commercial options

Different types of scans:

- **license** scanning
  - understand which licenses apply
- **dependency** scanning
  - identify interactions between components, projects
- **vulnerability** scanning
  - identify known versions with known problems (CVE, CWE)
  - source code analysis (static, dynamic)
License Scanning Tools

**FOSSology**

Server based framework used to scan a codebase for licenses, copyright and export statements

Performs textual analysis and regular expression scanning to identify likely license notices and references

Manual review results to remove false positives and investigate unusual findings, can be retained between versions

https://www.fossology.org
License Scanning Tools

ScanCode

Command-line tool used to scan a codebase for licenses and copyright statements

Need to manually review output to remove false positives and investigate unusual findings

https://github.com/nexB/scancode-toolkit
License Scanning Tools

OSS Review Toolkit (ORT)

Analyzes the source code for dependencies, downloads the source code of the dependencies, scans all source code for license information, and summarizing results.

ORT is a suite of different tools that are designed as libraries (for programmatic use) with a minimal command line interface (for scripted use).

https://github.com/heremaps/oss-review-toolkit
License Scanning Tools

Quartermaster

Tooling to assist with open source compliance as part of a CI/CD toolchain, and analyzing the results of a project’s build process

Still in an early stage of development

Useful for integrating open source license compliance into your CI/CD infrastructure

https://qmstr.org
License Scanning Tools

**Tern**

Command-line tool to identify components and licenses installed by a container image or manifest

Still in an early stage of development

Useful for understanding licenses that apply to the dependencies (and sub-dependencies) when using containers

https://github.com/vmware/tern
License Scanning Tools

Keep in mind:

However automated the tooling is, some **manual review** will likely be required.
Communicating SBOM Information

**Goal 1:** Be accurate about software being used inside an organization

**Goal 2:** Be accurate about license information and versioning for products and patches distributed

**Goal 3:** Do so in an automated, scalable way
Communicating SBOM Information: SPDX

Used to communicate software identification, license and security information in standardized, machine-readable formats

SPDX files can be produced from source code scans or builds, curated and annotated by reviewers, and shared between organizations

Based on 8 years of analysis of use cases, incorporating input from industry experts in packaging, licensing and security

https://spdx.github.io/spdx-spec
SPDX Documents comprise manifests of files from software packages.

Includes checksum hashes per file, license information and other optional data.

Two official formats:
- RDF – easier for automated consumption
- Tag-value - easier for human consumption

Translation tools can convert to XML, JSON, YAML, spreadsheets, to your favorite report.

https://spdx.github.io/spdx-spec
Communicating SBOM Information: SPDX

SPDX Documents

FileName: /requirements.txt
SPDXID: SPDXRef-item3456870
FileChecksum: SHA1: 3fd8978ad3dfafaa5f...
LicenseConcluded: Apache-2.0
LicenseInfoInFile: Apache-2.0
FileCopyrightText: NONE

FileName: /README.md
SPDXID: SPDXRef-item3456871

https://spdx.github.io/spdx-spec
Communicating SBOM Information: SPDX

SPDX License List

From the License List:

“...a list of commonly found licenses and exceptions used in free and open source and other collaborative software or documentation.”

“The purpose of the SPDX License List is to enable easy and efficient identification of such licenses and exceptions in an SPDX document, in source files or elsewhere.”

https://spdx.org/licenses
Communicating SBOM Information: SPDX

SPDX License List

Examples:

- BSD-2-Clause
- BSD-3-Clause
- GPL-3.0-only
- GPL-3.0-or-later
- MIT
- MPL-2.0

https://spdx.org/licenses
Communicating SBOM Information: SPDX

SPDX Short-Form IDs

Usage example:

Examples:

// SPDX-License-Identifier: Apache-2.0

// SPDX-License-Identifier: GPL-2.0-only OR MIT

// SPDX-License-Identifier: Apache-2.0 AND MIT

https://spdx.org/ids
Communicating SBOM Information: SPDX

One line comment in each source code file to unambiguously designate the applicable license(s)

Examples:
// SPDX-License-Identifier: Apache-2.0

// SPDX-License-Identifier: GPL-2.0-only OR MIT

// SPDX-License-Identifier: Apache-2.0 AND MIT

https://spdx.org/ids

Usage example:

// SPDX-License-Identifier: GPL-2.0-only OR MIT

// SPDX-License-Identifier: Apache-2.0 AND MIT
License Compliance Three-way Handshake

Management

Legal

Engineering

Triquetra Cross Image: By Madboy74 - Own work, CC BY-SA 4.0
Resources: Licenses and Legal

SPDX License List: https://spdx.org/licenses

Open Source Initiative Approved Licenses: https://opensource.org/licenses

Free Software Foundation Comments on Licenses: https://www.gnu.org/licenses/license-list.en.html

Book: “Open (Source) for Business: A Practical Guide to Open Source Software Licensing” by Heather Meeker
Resources: License Compliance Processes

Whitepapers:

Open Source Compliance in the Enterprise:

License Scanning and Compliance Programs for FOSS Projects:
Resources: License Scanning

Open Source Tools:

- FOSSology: https://github.com/fossology
- ScanCode: https://github.com/nexB/scancode-toolkit/
- ORT: https://github.com/heremaps/oss-review-toolkit
- Tern: https://github.com/vmware/tern
- Quartermaster: https://github.com/qmstr
Open FinTech Forum

AI, Blockchain & Kubernetes on Wall Street