Open FinTech Forum

AI, Blockchain & Kubernetes on Wall Street

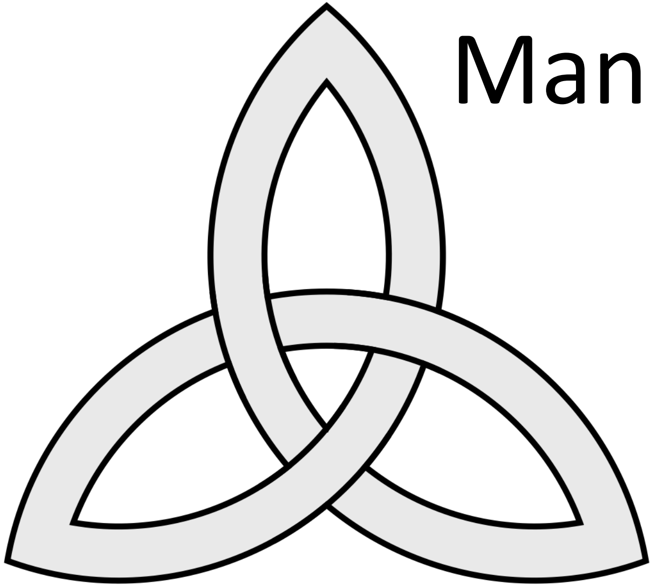
Complying with Open Source Licenses

Kate Stewart, Senior Director of Strategic Programs

Steve Winslow, Director of Strategic Programs

THE LINUX FOUNDATION

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)

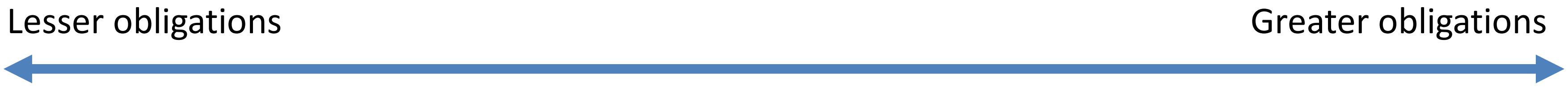
Lesser obligations

Greater obligations

Permissive
(aka “Attribution”)

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

Open Source License Types (broadly speaking)



Lesser obligations

Greater obligations

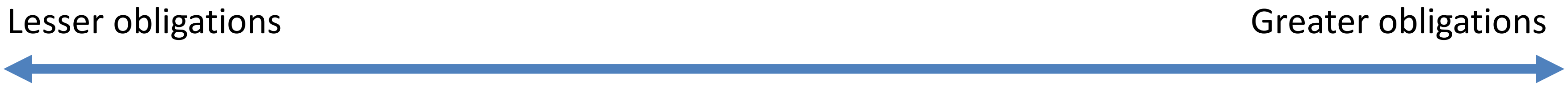
Permissive
(aka “Attribution”)

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

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

Greater 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.

Open Source License Types (broadly speaking)

Lesser obligations

Greater obligations

Permissive

Common examples:

- BSD-2-Clause
- BSD-3-Clause
- MIT
- Apache-2.0

Weak Copyleft

Common examples:

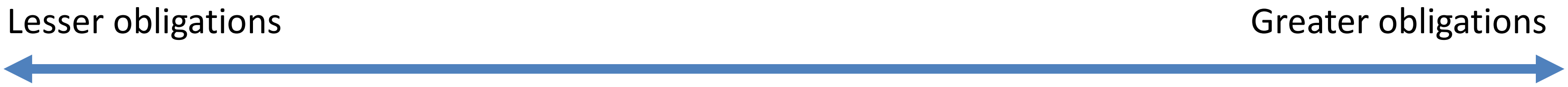
- GNU Lesser General Public License (LGPL)
- Mozilla Public License (MPL)
- Eclipse Public License (EPL)
- Common Development and Distribution License (CDDL)

Strong Copyleft

Common examples:

- GNU General Public License (GPL)
- GNU Affero GPL (AGPL)

Open Source License Types (broadly speaking)

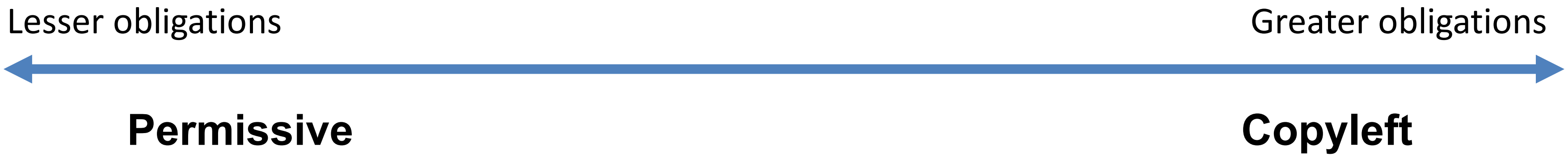


Lesser obligations

Greater obligations

Permissive		**Copyleft**
Creative Commons Attribution 4.0 (CC-BY-4.0)	**Documentation and Creative Content**	Creative Commons Attribution 4.0 ShareAlike (CC-BY-SA-4.0)
Community Data License Agreement – Permissive (CDLA-Permissive-1.0)	**Data**	Community Data License Agreement – Sharing (CDLA-Sharing-1.0)

Open Source License Types (broadly speaking)



Lesser obligations

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

Management

Managing the risk of using open source software

Management: Accurate Inventory

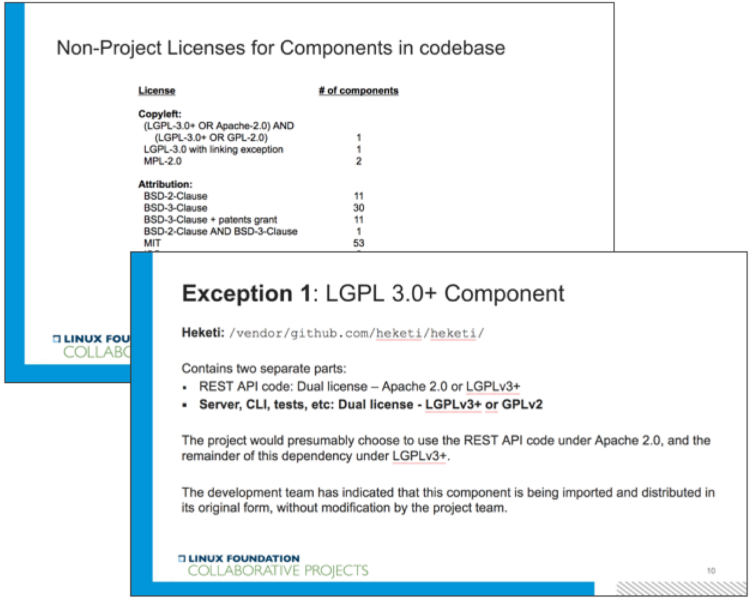
This is not an insurmountable challenge

- 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

Sharing SBOMs: What Info is Significant?



Non-Project Licenses for Components in codebase

License	# of components
Copyleft:	
(LGPL-3.0+ OR Apache-2.0) AND (LGPL-3.0+ OR GPL-2.0)	1
LGPL-3.0 with linking exception	1
MPL-2.0	2
Attribution:	
BSD-2-Clause	11
BSD-3-Clause	30
BSD-3-Clause + patents grant	11
BSD-2-Clause AND BSD-3-Clause	1
MIT	53

LINUX FOU
COLLABO

Exception 1: LGPL 3.0+ Component

Heketi: `/vendor/github.com/heketi/heketi/`

Contains two separate parts:

- REST API code: Dual license – Apache 2.0 or LGPLv3+
- **Server, CLI, tests, etc: Dual license - LGPLv3+ or GPLv2**

The project would presumably choose to use the REST API code under Apache 2.0, and the remainder of this dependency under LGPLv3+.

The development team has indicated that this component is being imported and distributed in its original form, without modification by the project team.

LINUX FOUNDATION
COLLABORATIVE PROJECTS

10

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)

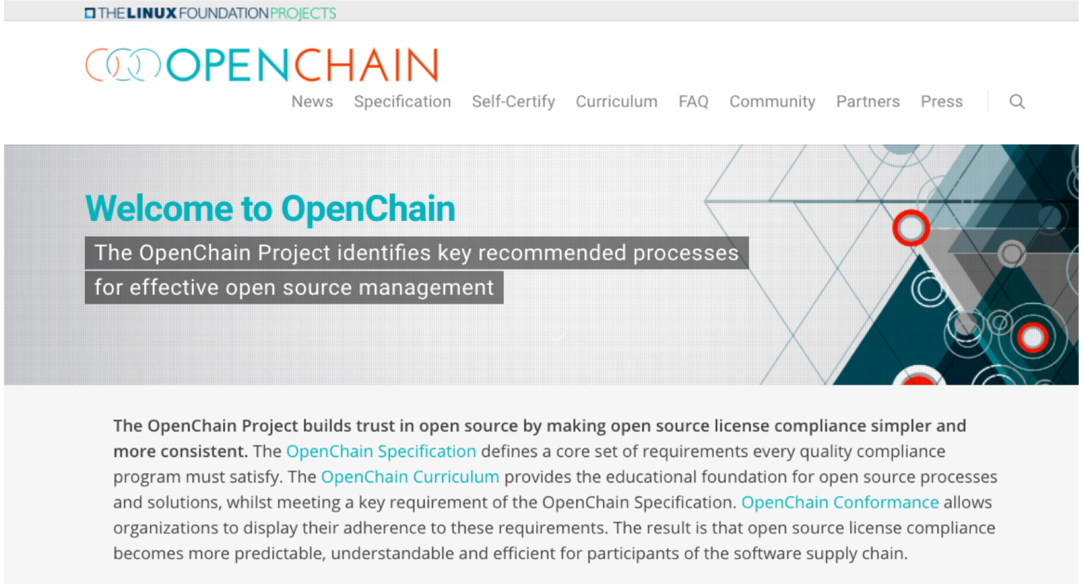
SParts (Distributed ledger for logging and tracking in supply chain)

REUSE (Artifact organization in projects)

Core Infrastructure Initiative (Security and Transparency)

Best Practices: Between Organizations

The **OpenChain Project** documents the processes to build trust between members of a software supply chain using open source software.



Welcome to OpenChain

The OpenChain Project identifies key recommended processes for effective open source management

The OpenChain Project builds trust in open source by making open source license compliance simpler and more consistent. The OpenChain Specification defines a core set of requirements every quality compliance program must satisfy. The OpenChain Curriculum provides the educational foundation for open source processes and solutions, whilst meeting a key requirement of the OpenChain Specification. OpenChain Conformance allows organizations to display their adherence to these requirements. The result is that open source license compliance becomes more predictable, understandable and efficient for participants of the software supply chain.

Best Practices: Between Organizations

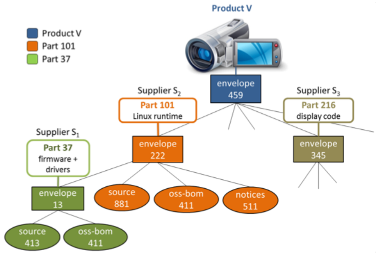
SParts (Software Parts Ledger) uses blockchain technology to manage open source across the supply chain.

Utilizes Hyperledger Sawtooth Platform and SPDX-based SBOM to conform to OpenChain best practices.

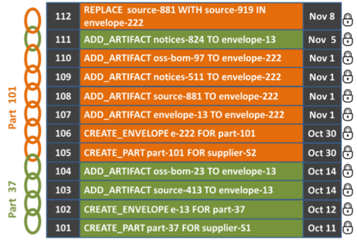
See: https://github.com/Wind-River/sparts

Accepted into Hyperledger Labs, Mar. 2018: https://github.com/hyperledger-labs/hyperledger-labs.github.io/blob/master/labs/SParts.md

SParts
Software Parts Ledger



Software Parts Ledger


Part 101
Part 37

112	REPLACE source-881 WITH source-919 IN envelope-222	Nov 8
111	ADD_ARTIFACT notices-824 TO envelope-13	Nov 5
110	ADD_ARTIFACT oss-bom-97 TO envelope-222	Nov 1
109	ADD_ARTIFACT notices-511 TO envelope-222	Nov 1
108	ADD_ARTIFACT source-881 TO envelope-222	Nov 1
107	ADD_ARTIFACT envelope-13 TO envelope-222	Nov 1
106	CREATE_ENVELOPE e-222 FOR part-101	Oct 30
105	CREATE_PART part-101 FOR supplier-S2	Oct 30
104	ADD_ARTIFACT oss-bom-23 TO envelope-13	Oct 14
103	ADD_ARTIFACT source-413 TO envelope-13	Oct 14
102	CREATE_ENVELOPE e-13 FOR part-37	Oct 12
101	CREATE_PART part-37 FOR supplier-S1	Oct 11

Best Practices: In Projects

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/

Practices: https://reuse.software/practices/2.0/

Overview: https://reuse.software/reuse/reuse-presentation.pdf

REUSE compliant

REUSE SOFTWARE

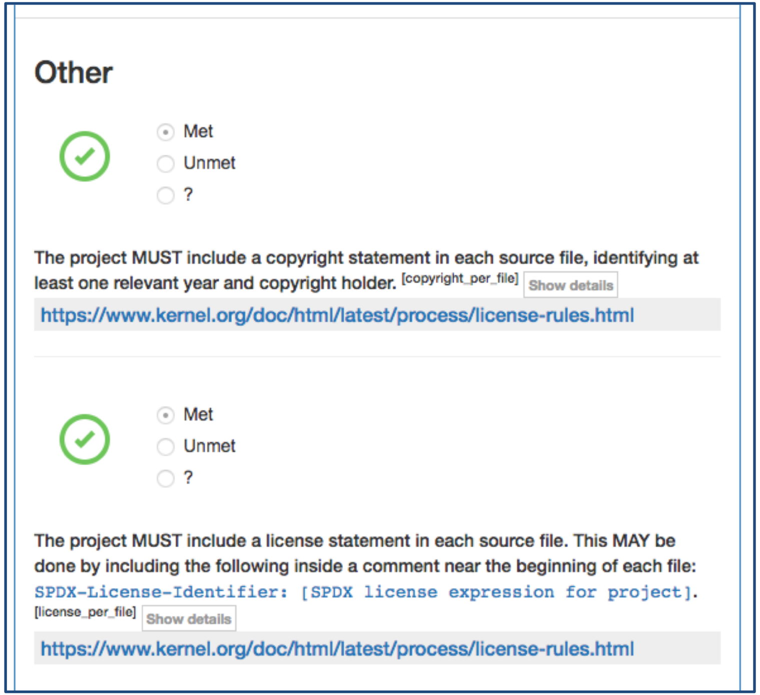
Best practices for license information in ways not only humans can read, but *computers* as well. Machine readable copyright and license information, simply put!

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.:
https://bestpractices.coreinfrastructure.org/en/projects/34?criteria_level=2



Other

✓ ● Met ○ Unmet ○ ?

The project MUST include a copyright statement in each source file, identifying at least one relevant year and copyright holder. [copyright_per_file] Show details

https://www.kernel.org/doc/html/latest/process/license-rules.html

✓ ● Met ○ Unmet ○ ?

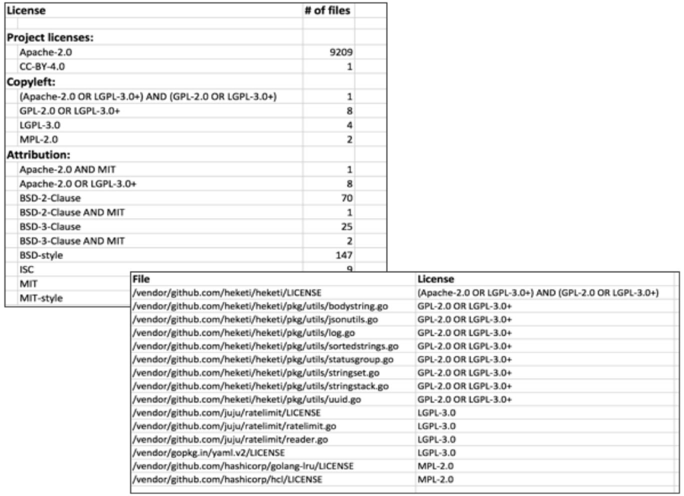
The project MUST include a license statement in each source file. This MAY be done by including the following inside a comment near the beginning of each file: `SPDX-License-Identifier: [SPDX license expression for project]`. [license_per_file] Show details

https://www.kernel.org/doc/html/latest/process/license-rules.html

Engineering

Figuring out what open source software you have...

Sharing SBOMs: What Info is Significant?



License	# of files
Project licenses:	
Apache-2.0	9209
CC-BY-4.0	1
Copyleft:	
(Apache-2.0 OR LGPL-3.0+) AND (GPL-2.0 OR LGPL-3.0+)	1
GPL-2.0 OR LGPL-3.0+	8
LGPL-3.0	4
MPL-2.0	2
Attribution:	
Apache-2.0 AND MIT	1
Apache-2.0 OR LGPL-3.0+	8
BSD-2-Clause	70
BSD-2-Clause AND MIT	1
BSD-3-Clause	25
BSD-3-Clause AND MIT	2
BSD-style	147
ISC	9
MIT	
MIT-style	

File	License
/vendor/github.com/heketi/heketi/LICENSE	(Apache-2.0 OR LGPL-3.0+) AND (GPL-2.0 OR LGPL-3.0+)
/vendor/github.com/heketi/heketi/pkg/utils/bodystring.go	GPL-2.0 OR LGPL-3.0+
/vendor/github.com/heketi/heketi/pkg/utils/jsonutils.go	GPL-2.0 OR LGPL-3.0+
/vendor/github.com/heketi/heketi/pkg/utils/log.go	GPL-2.0 OR LGPL-3.0+
/vendor/github.com/heketi/heketi/pkg/utils/sortedstrings.go	GPL-2.0 OR LGPL-3.0+
/vendor/github.com/heketi/heketi/pkg/utils/statusgroup.go	GPL-2.0 OR LGPL-3.0+
/vendor/github.com/heketi/heketi/pkg/utils/stringset.go	GPL-2.0 OR LGPL-3.0+
/vendor/github.com/heketi/heketi/pkg/utils/stringstack.go	GPL-2.0 OR LGPL-3.0+
/vendor/github.com/heketi/heketi/pkg/utils/uuid.go	GPL-2.0 OR LGPL-3.0+
/vendor/github.com/juju/ratelimit/LICENSE	LGPL-3.0
/vendor/github.com/juju/ratelimit/ratelimit.go	LGPL-3.0
/vendor/github.com/juju/ratelimit/reader.go	LGPL-3.0
/vendor/gopkg.in/yaml.v2/LICENSE	LGPL-3.0
/vendor/github.com/hashicorp/golang-lru/LICENSE	MPL-2.0
/vendor/github.com/hashicorp/hcl/LICENSE	MPL-2.0

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

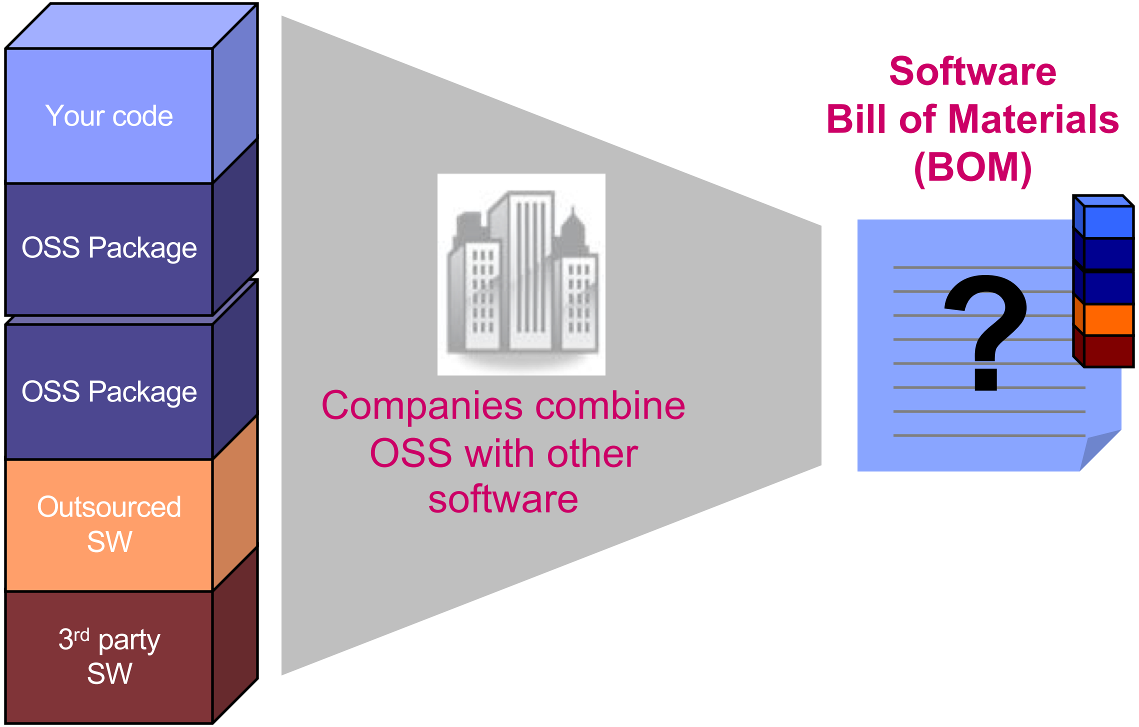
Typical high-level process:

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?



Your code

OSS Package

OSS Package

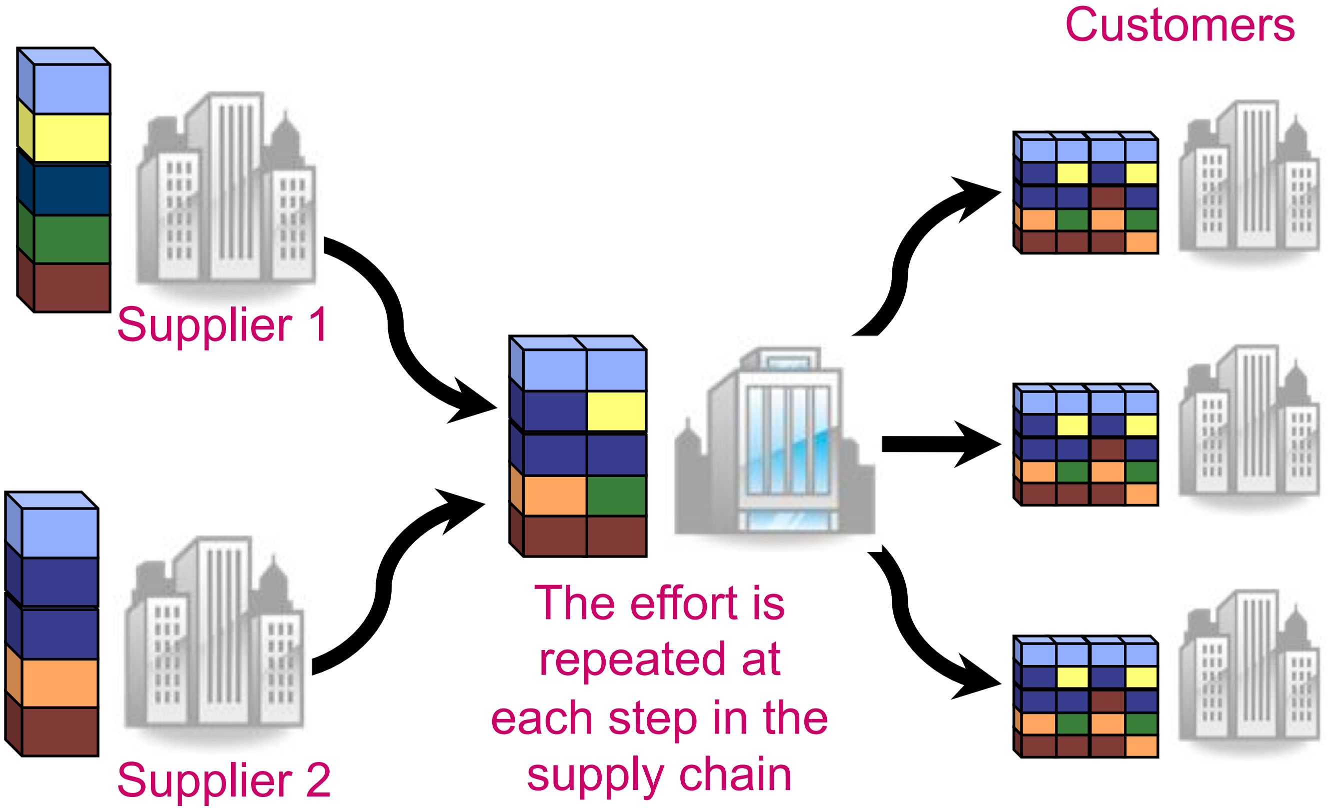
Outsourced SW

3rd party SW

Companies combine OSS with other software

Software Bill of Materials (BOM)

Supply Chain Impact



Customers

Supplier 1

Supplier 2

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

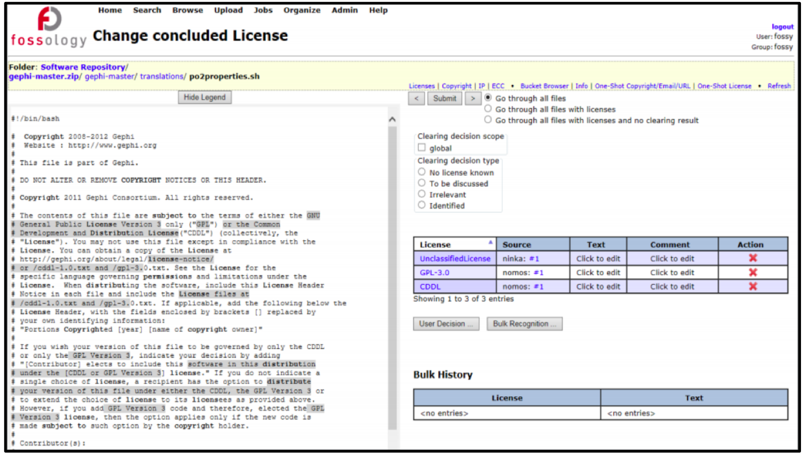
Various scanning tools and services exist, including open source and proprietary / commercial options

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


Home
Search
Browse
Upload
Jobs
Organize
Admin
Help
fossology
logout
Licenses | Copyright | IP | ECC • Bucket Browser | Info | One-Shot Copyright/Email/URL | One-Shot License • Refresh
Hide Legend
<
Submit
>
User Decision ...
Bulk Recognition ...

Change concluded License

User: fossy
Group: fossy

Folder: Software Repository/
gephi-master.zip/ gephi-master/ translations/ po2properties.sh

```
#!/bin/bash

#  Copyright 2008-2012 Gephi
#  Website : http://www.gephi.org
#
# This file is part of Gephi.
#
# DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
#
# Copyright 2011 Gephi Consortium. All rights reserved.
#
# The contents of this file are subject to the terms of either the GNU
# General Public License Version 3 only ("GPL") or the Common
# Development and Distribution License("CDDL") (collectively, the
# "License"). You may not use this file except in compliance with the
# License. You can obtain a copy of the License at
# http://gephi.org/about/legal/license-notice/
# or /cddl-1.0.txt and /gpl-3.0.txt. See the License for the
# specific language governing permissions and limitations under the
# License.  When distributing the software, include this License Header
# Notice in each file and include the License files at
# /cddl-1.0.txt and /gpl-3.0.txt. If applicable, add the following below the
# License Header, with the fields enclosed by brackets [] replaced by
# your own identifying information:
# "Portions Copyrighted [year] [name of copyright owner]"
#
# If you wish your version of this file to be governed by only the CDDL
# or only the GPL Version 3, indicate your decision by adding
# "[Contributor] elects to include this software in this distribution
# under the [CDDL or GPL Version 3] license." If you do not indicate a
# single choice of license, a recipient has the option to distribute
# your version of this file under either the CDDL, the GPL Version 3 or
# to extend the choice of license to its licensees as provided above.
# However, if you add GPL Version 3 code and therefore, elected the GPL
# Version 3 license, then the option applies only if the new code is
# made subject to such option by the copyright holder.
#
# Contributor(s):
```

● Go through all files
○ Go through all files with licenses
○ Go through all files with licenses and no clearing result

Clearing decision scope
□ global

Clearing decision type
○ No license known
○ To be discussed
○ Irrelevant
○ Identified

License	Source	Text	Comment	Action
UnclassifiedLicense	ninka: #1	Click to edit	Click to edit	✗
GPL-3.0	nomos: #1	Click to edit	Click to edit	✗
CDDL	nomos: #1	Click to edit	Click to edit	✗

Showing 1 to 3 of 3 entries

Bulk History

License	Text
<no entries>	<no entries>

https://www.fossology.org

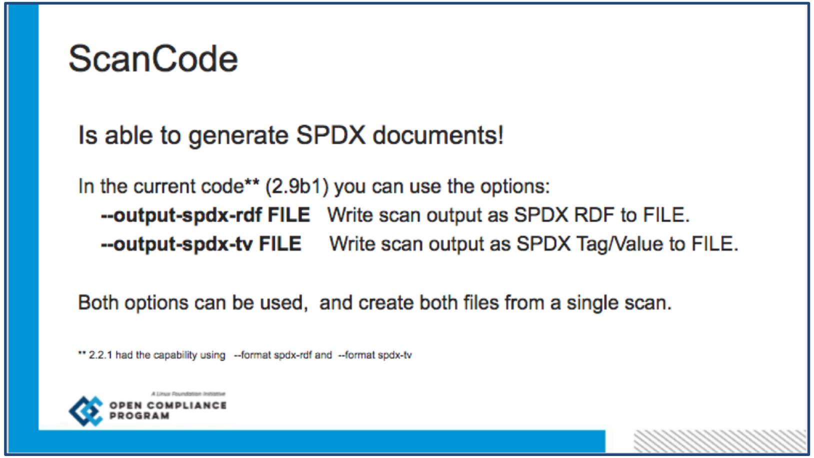
Server based framework used to scan a codebase for licenses, copyright and export statements

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

License Scanning Tools

ScanCode



ScanCode

Is able to generate SPDX documents!

In the current code** (2.9b1) you can use the options:

```
--output-spdx-rdf FILE   Write scan output as SPDX RDF to FILE.
--output-spdx-tv FILE    Write scan output as SPDX Tag/Value to FILE.
```

Both options can be used, and create both files from a single scan.

** 2.2.1 had the capability using --format spdx-rdf and --format spdx-tv

A Linux Foundation Initiative
OPEN COMPLIANCE PROGRAM

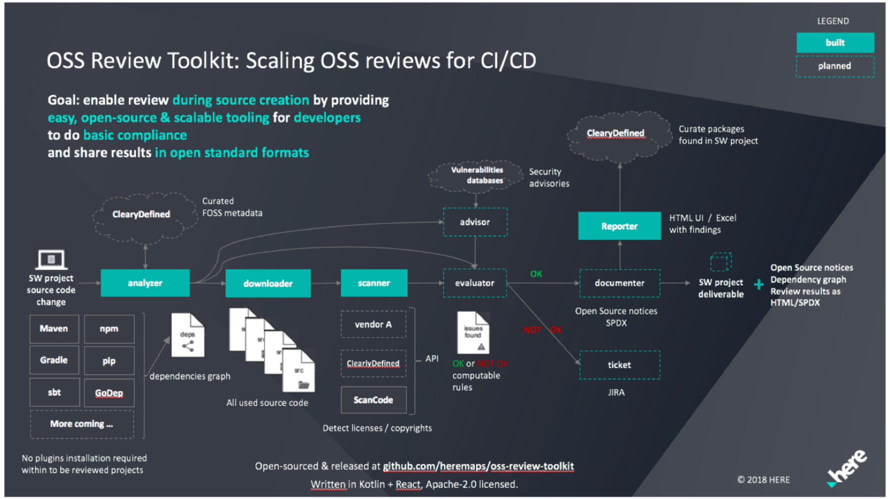
https://github.com/nexB/scancode-toolkit

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

License Scanning Tools

OSS Review Toolkit (ORT)



OSS Review Toolkit: Scaling OSS reviews for CI/CD

Goal: enable review during source creation by providing easy, open-source & scalable tooling for developers to do basic compliance and share results in open standard formats

Open-sourced & released at github.com/heremaps/oss-review-toolkit

Written in Kotlin + React, Apache-2.0 licensed.

© 2018 HERE

https://github.com/heremaps/oss-review-toolkit

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).

License Scanning Tools

Quartermaster

https://qmstr.org

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

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.

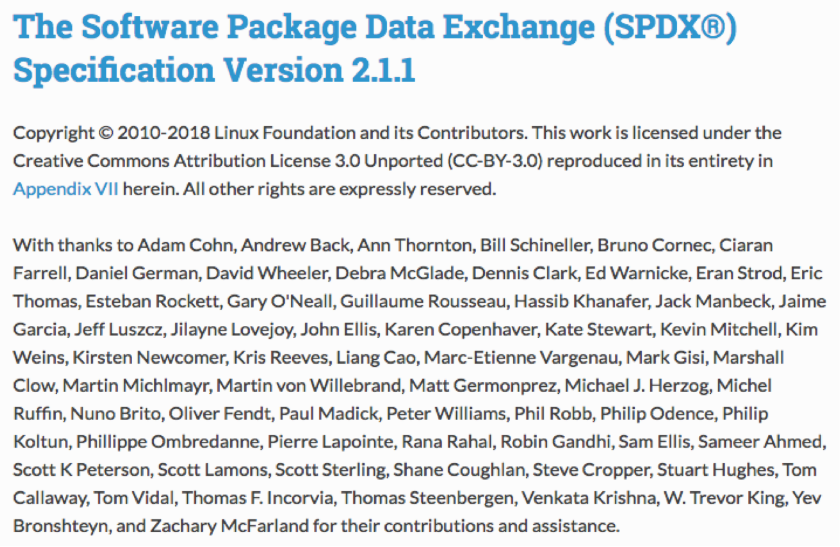
Communicating SBOM Information

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



The Software Package Data Exchange (SPDX®) Specification Version 2.1.1

With thanks to Adam Cohn, Andrew Back, Ann Thornton, Bill Schineller, Bruno Cornec, Ciaran Farrell, Daniel German, David Wheeler, Debra McGlade, Dennis Clark, Ed Warnicke, Eran Strod, Eric Thomas, Esteban Rockett, Gary O'Neall, Guillaume Rousseau, Hassib Khanafer, Jack Manbeck, Jaime Garcia, Jeff Luszcz, Jilayne Lovejoy, John Ellis, Karen Copenhaver, Kate Stewart, Kevin Mitchell, Kim Weins, Kirsten Newcomer, Kris Reeves, Liang Cao, Marc-Etienne Vargenau, Mark Gisi, Marshall Clow, Martin Michlmayr, Martin von Willebrand, Matt Germonprez, Michael J. Herzog, Michel Ruffin, Nuno Brito, Oliver Fendt, Paul Madick, Peter Williams, Phil Robb, Philip Odence, Philip Koltun, Phillippe Ombredanne, Pierre Lapointe, Rana Rahal, Robin Gandhi, Sam Ellis, Sameer Ahmed, Scott K Peterson, Scott Lamons, Scott Sterling, Shane Coughlan, Steve Cropper, Stuart Hughes, Tom Callaway, Tom Vidal, Thomas F. Incorvia, Thomas Steenbergen, Venkata Krishna, W. Trevor King, Yev Bronshteyn, and Zachary McFarland for their contributions and assistance.

https://spdx.github.io/spdx-spec

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

Communicating SBOM Information: SPDX

SPDX Documents

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.

Communicating SBOM Information: SPDX

SPDX

```
##File

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

##File

FileName: /README.md
SPDXID: SPDXRef-item3456871
...
```

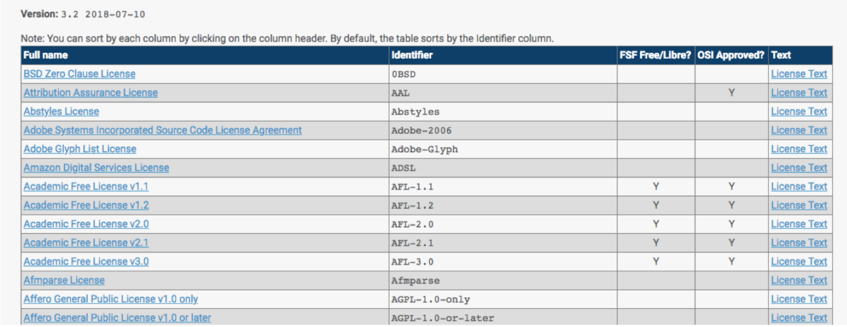
SPDX Documents

https://spdx.github.io/spdx-spec

Communicating SBOM Information: SPDX

SPDX

SPDX License List



Version: 3.2 2018-07-10

Note: You can sort by each column by clicking on the column header. By default, the table sorts by the Identifier column.

Full name	Identifier	FSF Free/Libre?	OSI Approved?	Text
BSD Zero Clause License	0BSD			License Text
Attribution Assurance License	AAL		Y	License Text
Abstyles License	Abstyles			License Text
Adobe Systems Incorporated Source Code License Agreement	Adobe-2006			License Text
Adobe Glyph List License	Adobe-Glyph			License Text
Amazon Digital Services License	ADSL			License Text
Academic Free License v1.1	AFL-1.1	Y	Y	License Text
Academic Free License v1.2	AFL-1.2	Y	Y	License Text
Academic Free License v2.0	AFL-2.0	Y	Y	License Text
Academic Free License v2.1	AFL-2.1	Y	Y	License Text
Academic Free License v3.0	AFL-3.0	Y	Y	License Text
Afmparse License	Afmparse			License Text
Affero General Public License v1.0 only	AGPL-1.0-only			License Text
Affero General Public License v1.0 or later	AGPL-1.0-or-later			License Text

https://spdx.org/licenses

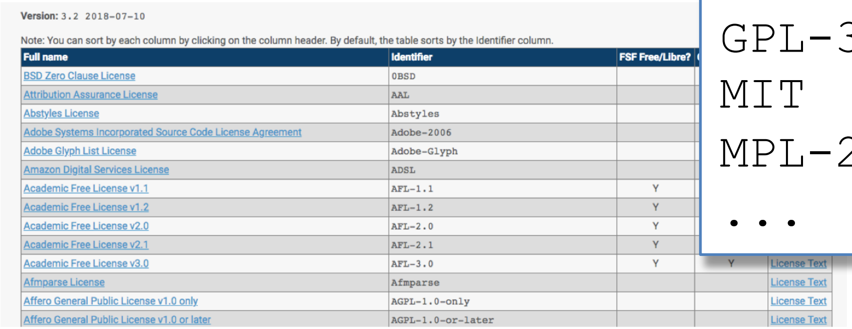
From the License List:

"...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."

Communicating SBOM Information: SPDX

SPDX License List



Version: 3.2 2018-07-10

Note: You can sort by each column by clicking on the column header. By default, the table sorts by the Identifier column.

Full name	Identifier	FSF Free/Libre?		
BSD Zero Clause License	0BSD			
Attribution Assurance License	AAL			
Abstyles License	Abstyles			
Adobe Systems Incorporated Source Code License Agreement	Adobe-2006			
Adobe Glyph List License	Adobe-Glyph			
Amazon Digital Services License	ADSL			
Academic Free License v1.1	AFL-1.1	Y		
Academic Free License v1.2	AFL-1.2	Y		
Academic Free License v2.0	AFL-2.0	Y		
Academic Free License v2.1	AFL-2.1	Y		
Academic Free License v3.0	AFL-3.0	Y	Y	License Text
Afmparse License	Afmparse			License Text
Affero General Public License v1.0 only	AGPL-1.0-only			License Text
Affero General Public License v1.0 or later	AGPL-1.0-or-later			License Text

https://spdx.org/licenses

```
Examples:

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

se List:

nonly found licenses
used in free and
d other collaborative
umentation.”

the SPDX License
easy and efficient
identification of such licenses and exceptions in an SPDX document, in source files or elsewhere.”

Communicating SBOM Information: SPDX

SPDX

SPDX Short-Form IDs

Usage example:
https://www.kernel.org/doc/html/latest/process/license-rules.html

https://spdx.org/ids

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
```

Communicating SBOM Information: SPDX

SPDX

and if a file’s license ID would look like this, maybe **rethink** that file’s structure:

```
GPL-3.0 AND GPL-2.0+ AND GPL-2.0 AND LGPL-2.1+ AND LGPL-2.1
AND MIT AND BSD-3-Clause AND (AFL-2.1+ OR BSD-3-Clause) AND
(MIT OR LicenseRef-BSD OR LicenseRef-GPL) AND (MIT OR
LicenseRef-GPL) AND (MPL-1.1 OR GPL-2.0 OR LGPL-2.1) AND
LicenseRef-MIT-style
```

Usage example:

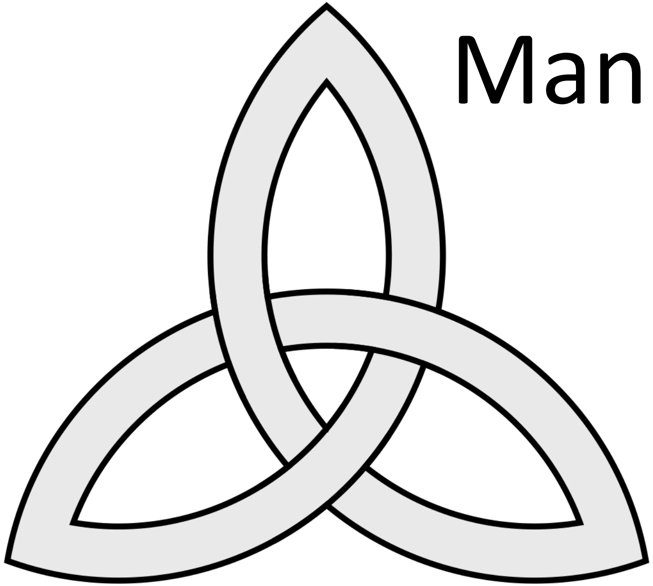
https://www.kernel.org/doc/html/latest/process/license-rules.html

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

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

https://spdx.org/ids

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:

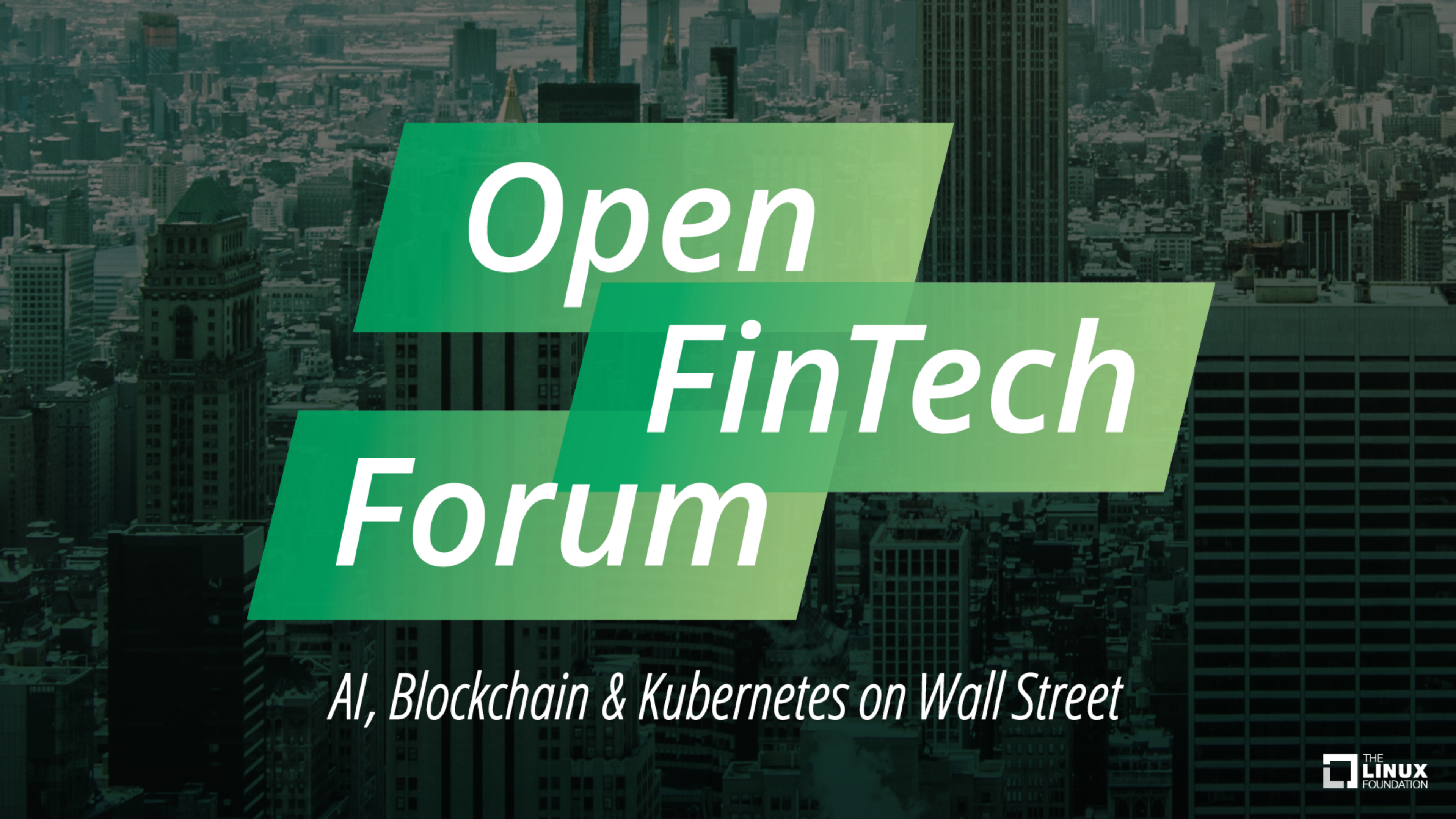
Open Source Compliance in the Enterprise:
https://www.linuxfoundation.org/open-source-management/2016/11/open-source-compliance-enterprise/

License Scanning and Compliance Programs for FOSS Projects:
https://www.linuxfoundation.org/publications/license-scanning-compliance-programs-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

THE LINUX FOUNDATION