

OLF *Live*

MENTORSHIP SERIES



Open Source Licensing

Steve Winslow, Director of Strategic Programs
The Linux Foundation

Agenda:

1. Welcome and Intro
2. What does “Open Source” mean?
3. Copyrights, Patents, Trademarks
4. Open Source License Types
5. Managing License Information
6. Contributing to Projects

Links and Resources for additional information available at end of slides

Contact me!

Steve Winslow

swinslow@linuxfoundation.org

GitHub: swinslow

I'm a FOSS coder and a FOSS lawyer

(I'm not your lawyer)

(Don't treat this as legal advice)

Agenda:

1. Welcome and Intro
2. What does “Open Source” mean?
3. Copyrights, Patents, Trademarks
4. Open Source License Types
5. Managing License Information
6. Contributing to Projects

What does “Open Source” mean?

Is it “software I can download for free”?

Is it “source code I can download for free”?

Is it “source code I can contribute back to”?

What are “Open Source Licenses”?

Informally and in a nutshell:

Legal terms (can be one sentence or several pages long)

which **grant broad rights** to **use, modify, and distribute** software

in both **source form** and **binary form**

typically written for use with **any type of software**

typically **standardized** (sort of — small set of most commonly-used licenses)

What are “Open Source Licenses”?

Informally and in a nutshell:

Typically impose

**responsibilities or conditions,
not restrictions,**

on redistribution or similar actions

“IF you redistribute, THEN you must do the following...”

as opposed to: *“You may not redistribute...”*

Example: MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), ***to deal in the Software without restriction***, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, ***subject to the following conditions***:

Example: MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), ***to deal in the Software without restriction***, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, ***subject to the following conditions***:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Example: MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), ***to deal in the Software without restriction***, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, ***subject to the following conditions***:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

[paragraph of ALL CAPS LEGALESE]

Formal definitions

Open Source Initiative's **Open Source Definition**



“open source”

opensource.org/osd-annotated

Free Software Foundation's **Four Essential Freedoms**



“free software”

gnu.org/philosophy/free-sw.html

Agenda:

1. Welcome and Intro
2. What does “Open Source” mean?
3. Copyrights, Patents, Trademarks
4. Open Source License Types
5. Managing License Information
6. Contributing to Projects

Copyright

Legal rights to control an original **creative work**, granted by each country's laws

In the US, includes the exclusive right to do the following: (see [17 U.S.C. § 106](#))

- **reproduce** the work
- prepare **derivative works** based on it
- **distribute** copies of it
- publicly **perform** and publicly **display** it

Exists automatically when an original creative work is fixed in a tangible medium

Registration with the US Copyright Office provides owner with additional rights

Patents

Legal rights to control an **invention**, granted by each country's laws

In the US, includes the exclusive right to do the following: (see [35 U.S.C. § 271\(a\)](#))

- **make** the invention
- **use** the invention
- **sell** or **offer to sell** the invention
- **import** the invention into the US

The invention must be useful, novel and non-obvious

Does not exist automatically; must apply with the US Patent and Trademark Office

Trademarks

Legal rights to control a **mark** — a name or logo designating the origin of goods or services

In the US, can apply to register a trademark with the Patent and Trademark Office

In the US, can also obtain rights just through use of the trademark

Typically not seen as explicitly included in most open source license grants, beyond certain circumstances

A complicated topic; we won't cover in detail here

Trade secrets

Legal rights to control **information** that:

1. the owner takes reasonable measures to keep secret;
2. where the information has economic value from not being generally known

Essentially the opposite of “open source”; we won’t cover in detail here

Example: MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), ***to deal in the Software without restriction***, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, ***subject to the following conditions***:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

[paragraph of ALL CAPS LEGALESE]

Agenda:

1. Welcome and Intro
2. What does “Open Source” mean?
3. Copyrights, Patents, Trademarks
4. Open Source License Types
5. Managing License Information
6. Contributing to Projects



Open Source License Types (informally, and from one perspective)

Lesser obligations

Greater obligations



Open Source License Types (informally, and from one perspective)

Lesser obligations

Greater obligations



Permissive

Main responsibility:

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

Open Source License Types (informally, and from one perspective)

Lesser obligations

Greater obligations



Permissive

Strong Copyleft

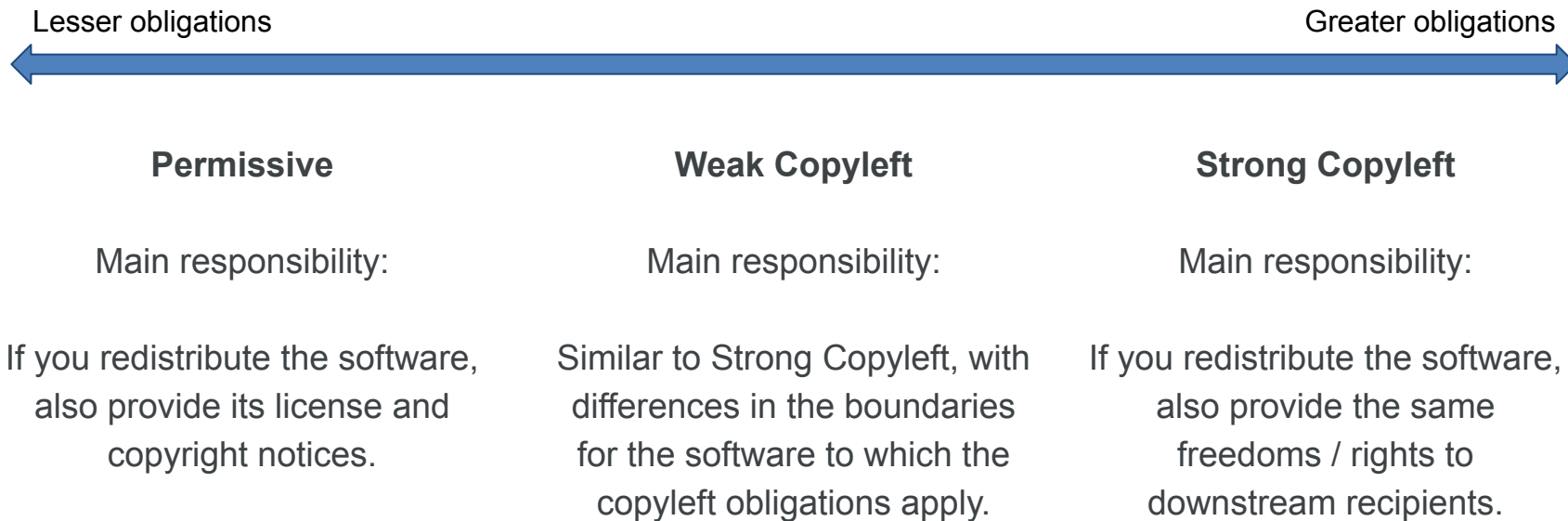
Main responsibility:

Main responsibility:

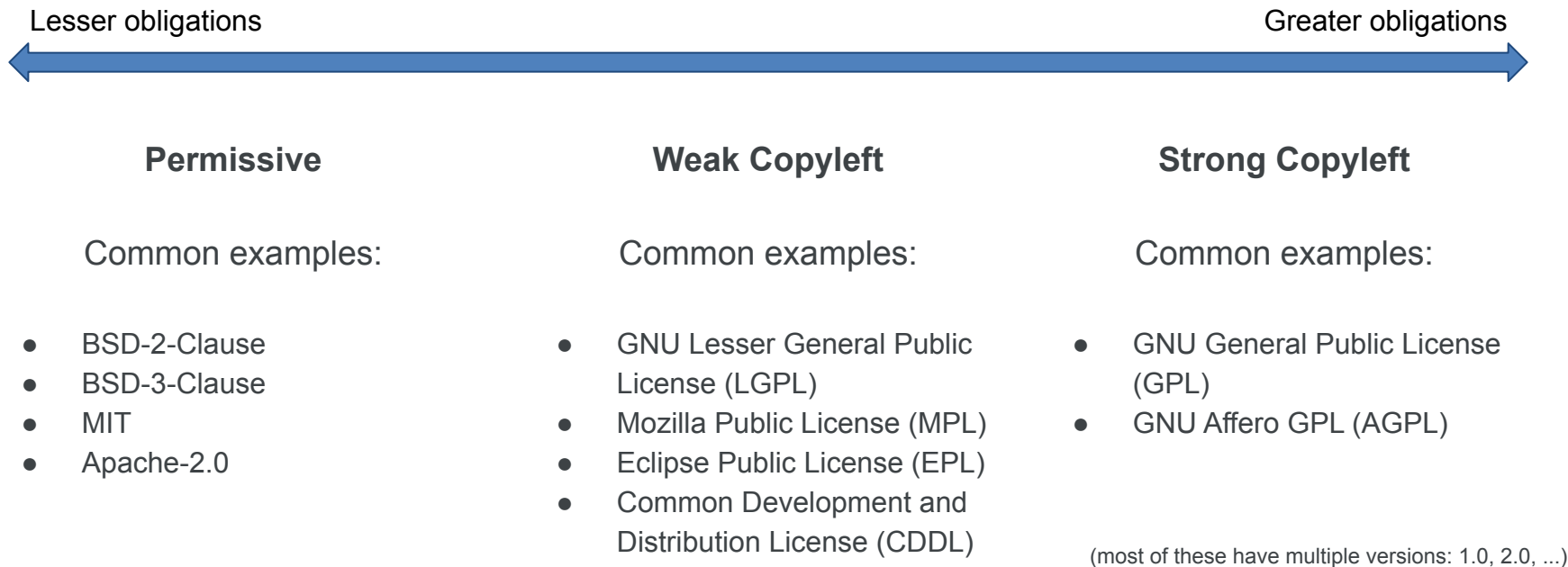
If you redistribute the software,
also provide its license and
copyright notices.

If you redistribute the software,
also provide the same
freedoms / rights to
downstream recipients.

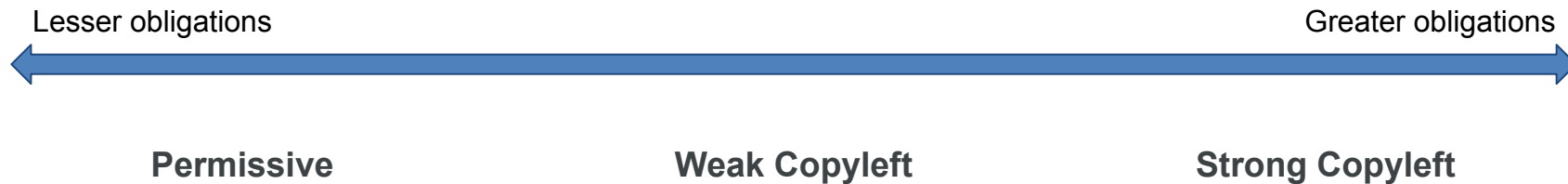
Open Source License Types (informally, and from one perspective)



Open Source License Types (informally, and from one perspective)



Open Source License Types (informally, and from one perspective)



There are more considerations than just these

But to get started with understanding open source licenses,
this spectrum is a useful framework for approaching them

Things other than software

The standard open source software licenses are drafted with software primarily in mind

There are other licenses that focus more on other types of content, such as:

- open documentation and specifications
- other creative written and artistic content
- open data
- open hardware

“Source Available”

Some licenses or license-like statements are not “free software” or “open source” licenses

If a project or organization makes source code available under terms that don’t meet these definitions, it is often referred to as a “source available” license

Agenda:

1. Welcome and Intro
2. What does “Open Source” mean?
3. Copyrights, Patents, Trademarks
4. Open Source License Types
5. Managing License Information
6. Contributing to Projects

What License(s) are relevant to a project?


- Its own primary license
- Different licenses for files or portions of files within the project
- Different licenses for external dependencies (and their subdependencies...)
- Potentially more, depending on the particular technology and scope of use

This can quickly become a large number of licenses

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

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
- The following slides focus on these two
- 



Identifying Licenses

Challenges:

- scale
- difficulty
- not fun

Identifying Licenses

For smaller projects, may be feasible to manage manually

For larger projects, hard to do at scale without automation / tools

There are various open source and proprietary tools and services that can assist with this

Pick one, understand what it does, and incorporate it into your development process

Communicating License Information

Once you know what licenses you've got,
communicate them in a standardized way

SPDX is:

“...an open standard for communicating software
bill of material information, including components,
licenses, copyrights, and security references.”



Software Package Data Exchange

<https://spdx.dev>

Communicating License Information

SPDX License List

<https://spdx.org/licenses>

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

Version: 3.11 2020-11-25

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?
BSD Zero Clause License	0BSD		Y
Attribution Assurance License	AAL		Y
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	Y
Academic Free License v1.2	AFL-1.2	Y	Y
Academic Free License v2.0	AFL-2.0	Y	Y
Academic Free License v2.1	AFL-2.1	Y	Y
Academic Free License v3.0	AFL-3.0	Y	Y
Afmmparse License	Afmmparse		
Affero General Public License v1.0 only	AGPL-1.0-only		
Affero General Public License v1.0 or later	AGPL-1.0-or-later		
GNU Affero General Public License v3.0 only	AGPL-3.0-only	Y	Y
GNU Affero General Public License v3.0 or later	AGPL-3.0-or-later	Y	Y
Aladdin Free Public License	Aladdin		
AMD's open source License	AMDSDPL		

Communicating License Information

SPDX short-form IDs

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

```
// 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.dev/ids>



<https://reuse.software>

Agenda:

1. Welcome and Intro
2. What does “Open Source” mean?
3. Copyrights, Patents, Trademarks
4. Open Source License Types
5. Managing License Information
6. Contributing to Projects

Contributing to Projects

Depending on the project, its “outbound” license might not be the same as its “inbound” license

Many projects, often smaller ones, accept contributions without addressing this at all

Larger projects often use a more formal contribution mechanism, typically:

- the Developer’s Certificate of Origin (DCO); and/or
- a Contributor License Agreement (CLA)

Developer's Certificate of Origin (DCO)

One text, used across many projects: <https://developercertificate.org/>

Basically an assertion that the contributor has the right to contribute the code they are offering, under the specified license

Typically asserted at the time of contribution, via a sign-off in the commit message:

```
Signed-off-by: Steve Winslow <steve@example.com>
```

Contributor License Agreement (CLA)

A legal agreement, signed by individuals and organizations before they can contribute

Many different kinds; they can have wildly different texts and effects

Read the CLA text (and discuss with your lawyer) to understand its effects before contributing

QLF *Live* MENTORSHIP
SERIES

Q&A

QLF *Live* MENTORSHIP
SERIES

Resources

Linux Foundation Training and Resources

Free Training Course: [Open Source Licensing Basics for Software Developers](#) (LFC191)

Blog posts and Best Practices papers, e.g.:

- [Copyright Notices in Open Source Software Projects](#)
- [Docker Containers and License Compliance](#)
- [Practical GPL Compliance](#)
- [Understanding Open Source Technology and US Export Controls](#)
- [Summary of GDPR Concepts for Open Source Projects](#)
- [Guide to Open Source Software for Procurement Professionals](#)

TODO Group [Open Source Guides](#)



LF Live MENTORSHIP SERIES

Linux Foundation Compliance Projects



defines the key requirements for an organization's open source compliance program

<https://www.openchainproject.org>



provides a forum for Open Source Program Officers to collaborate on best practices

<https://todogroup.org>



defines a specification for communicating software bill-of-material information

<https://spdx.dev>



Automating Compliance Tooling

supports development of open source tooling to facilitate compliance

<https://automatecompliance.org>

Open Source Compliance Tools

FOSSology

In-depth scans to identify license and copyright notices

<https://www.fossology.org>



OSS Review Toolkit (ORT)

Customizable pipeline of tools for open source license detection and review

<https://oss-review-toolkit.org>



Tern

Scanning for licenses in containers and container image dependencies

<https://github.com/tern-tools/tern>



ScanCode

One-shot, command line tool to scan for license and copyright notices

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



Thank you for joining us today!

We hope it will be helpful in your journey to learning more about effective and productive participation in open source projects. We will leave you with a few additional resources for your continued learning:

- The [LF Mentoring Program](#) is designed to help new developers with necessary skills and resources to experiment, learn and contribute effectively to open source communities.
- [Outreachy remote internships program](#) supports diversity in open source and free software
- [Linux Foundation Training](#) offers a wide range of [free courses](#), webinars, tutorials and publications to help you explore the open source technology landscape.
- [Linux Foundation Events](#) also provide educational content across a range of skill levels and topics, as well as the chance to meet others in the community, to collaborate, exchange ideas, expand job opportunities and more. You can find all events at events.linuxfoundation.org.