KubeCon | CloudNativeCon

North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

KubeCon | CloudNativeCon
North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Introduction to Open Source Licenses

BoF / Q&A

Jeff Shapiro, The Linux Foundation

Introduction

Objective

- Teach you about open source licenses and why they matter.

Key Results

- Understand license types & attributes.
- Identify license conflicts.
- Choose an open source license.

Disclaimer: I am an engineer, not a lawyer.

What is a license?

> *“A license defines the rights and obligations that a licensor grants to a licensee. Open source licenses grant licensees the right to copy, modify and redistribute source code. These licenses may also impose obligations...”*
>
> \- Wikipedia

What is a license?

Rights

- Use, inspect, and copy the source code.
- Make modifications.
- Create derivative works and redistribute.

Obligations & Conditions

Examples

- Provide attribution.
- Contribute changes.
- Provide derivative works under the same license.
- Disclaim warranty and liability.

License Types

Permissive

- Allows use for most situations.
- Typically few obligations.

Copyleft

- Ensures code* will remain open source.
- Less compatible with proprietary code.
- Strong copyleft vs. weak copyleft.

*including most derivative works

License Types

Public Domain

- No copyright, not strictly a license.
- Can be used for anything.
- May vary in different regions.

Proprietary

- Typical for commercial code.
- Source* may or may not be disclosed.
- Examples: EULA, ToS

*source code being disclosed does not automatically make it open source!

License Types - Examples

Permissive

- BSD, MIT, Apache, Artistic, CC-BY

Weak Copyleft

- LGPL, EPL, MPL, CPL, GPL w/ exception

Strong Copyleft

- GPL, AGPL, CC-BY-SA

➢ Most of these licenses have multiple versions.

License Example: The 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.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright vs License

COPYRIGHT ≠ LICENSE

- A copyright typically exists whether or not a copyright notice is explicitly stated.
- The copyright may be owned by the author, organization, or company.
- The copyright owner decides which license, if any, applies.*
- If you can't identify a license, the code might **not** be open source.

*except, see next slide!

Contributor License Agreement (CLA)

> *“A Contributor License Agreement (CLA) defines the terms under which intellectual property has been contributed to a company or project, typically software under an open source license.”*
>
> \- Wikipedia

- For many open source projects, each author retains the copyright for the code they write, and grants a license to the code via a CLA. The CLA typically states that all code contributed may be distributed under the project’s open source license.

Developer Certificate of Origin (DCO)

Introduced by the Linux community, the DCO represents that you have the rights to contribute code to a project.

- Any code being contributed was either created by you, or you have the right to submit the code.
- DCO ≠ License
- DCO ≠ CLA
- https://developercertificate.org

THE LINUX FOUNDATION

SPDX License Identifier

Software Package Data Exchange (SPDX) is an open standard for SBOM communication.

- A unique identifier for open source licenses.
- Communicate license information in a simple, efficient, portable and machine-readable format.
- Add an SPDX License Identifier along with your copyright to the file header in **every** source file.

Examples

```
// SPDX-License-Identifier: MIT
/* SPDX-License-Identifier: MIT OR Apache-2.0 */
#  SPDX-License-Identifer: GPL-2.0-or-later
```

Dual Licenses

Some projects are released under more than one license

Examples

Perl: GPL or Artistic

- You may use the code under either license.

Qt: GPL or Commercial

- You may purchase a commercial license, otherwise the GPL license applies.

License Obligations

When do obligations take effect?

Distribution

- Most obligations triggered by distribution.*

Integration

- Fully integrated, dynamic linking, or separate process?

Modifications

- Have you made modifications to the code?

*notable exception: AGPL

License Conflicts

Mixing different open source licenses

- Many licenses are compatible, but not all.
- Do they have conflicting obligations?
- Check with OSI, Apache, and GNU/FSF.

- Apache 2.0 + GPLv3 => no conflict *
- Apache 1.1 + GPLv3 => CONFLICT
- Apache (any) + GPLv2 => CONFLICT

Examples

* combined code is GPLv3, this is **unidirectional**!

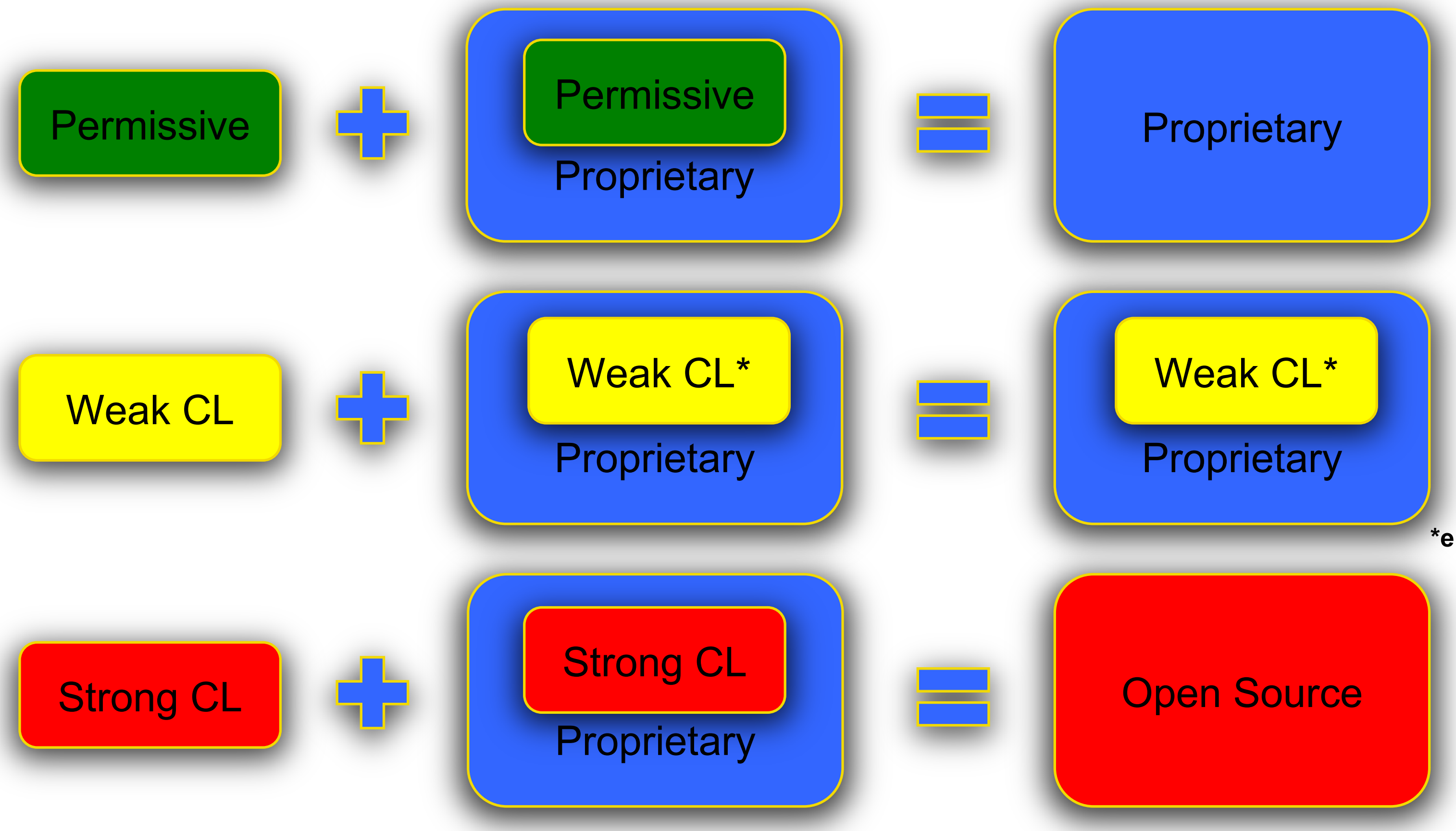
[see https://www.apache.org/licenses/GPL-compatibility.html]

License Conflicts

Mixing open source with proprietary code

- Most permissive licenses are compatible with proprietary code.
- Most copyleft licenses have some conflicts with proprietary licenses, **depending on how the code is integrated or combined**.

Mixing Open Source & Proprietary Code



Permissive + Permissive Proprietary = Proprietary

Weak CL + Weak CL* Proprietary = Weak CL* Proprietary

Strong CL + Strong CL Proprietary = Open Source

***e.g. dynamic linking**

Contributions

Contributing to an existing open source project

Project License

- Are you OK using the license for your code?
- What if there is no project license?

CLA, DCO, or ?

- Is there a CLA, DCO, or assignment of copyright, and what rights are you granting?
- Do you have the right to contribute the code?

Choosing a Project License

Who is your audience and what are your goals?

- Do you want to be proprietary code friendly?
- Are you OK with your code being distributed under a different license?
- Do you want to ensure that your code will always remain open source, including derivative works?

- Hundreds of licenses to choose from.
- Recommend: Choose a well-known license.

Choosing a Project License

Choosing a permissive license

- Proprietary code friendly.
- Typically few obligations, mainly attribution.
- Users can distribute combined works under a different license.
- Apache 2.0 includes an explicit patent license clause.

Choosing a Project License

Choosing a copyleft license

- Combined or derivative works typically must be licensed under the same license.
- May require modifications to be published or made available to the end user.
- May require user to allow reverse engineering for debugging purposes.
- Weak or strong copyleft depending on your goals.

THE LINUX FOUNDATION

License Resources

Open Source Initiative

- https://opensource.org

SPDX License List & ID

- https://spdx.org/licenses https://spdx.dev/ids

REUSE Software

- https://reuse.software

GNU/FSF License Comments

- https://www.gnu.org/licenses/license-list.en.html

Apache License Reference & FAQ

- https://www.apache.org/licenses/

LF Open Source Licensing Training Course

- https://training.linuxfoundation.org/training/open-source-licensing-basics-for-software-developers

BoF: Discussion and Q&A

- *What is your level of expertise in OSS licenses?*
- *Are you consuming or contributing code?*
- *Are you mixing open source and proprietary code?*
- *Are you choosing a project license?*
- *Do you need to understand a CLA or DCO?*
- *What is your individual use case?*

THE LINUX FOUNDATION