



Open Source Software Development

OSS Licensing & Legal Issues

Thái Minh Tuấn - Email: minhtuan@ctu.edu.vn

Learning Objectives

- What Are (Software) Licenses?
- Software License Types
- Open Source Definition
- OSS Restrictive (Copyleft/Protective) licenses
- OSS Permissive (Non-protective) licenses
- Software Patents
- Which License to Choose?
- Mix and Match License?
- What Is a Copyright?



What Are (Software) Licenses?

- *"A software license is a legal instrument . . . governing the use or redistribution of software"* (according to Wikipedia).
- **Software:**
 - A document that represents a program or firmware written by author(s)
 - Software program: A machine executable version of an algorithm or idea
- **Legal instrument:**
 - Formally expresses a legally enforceable act, process, or contractual duty, obligation, or right, and therefore evidences that act, process, or agreement
- **Governing the use:**
 - How a person or company can use this software program
 - Publicly post it, use inside their company, or personal use at home?
- **Redistribution:**
 - How it can be shared
 - Give a copy to a friend, modify or create derivative works, post online, etc.?

Software License Types

- Two broad categories: Proprietary or Open source
- Proprietary licenses:
 - Specific to a company or a project
 - Copyright holder keeps many or most of the rights to themselves
 - Often adds additional restrictions on what users can do
 - Commercial or freely available
- Open-source licenses:
 - Follow the definition of OSS developed by the Open Source Initiative
 - Allow software to be freely used, modified, and shared
 - Permissive or Copyleft
 - Patent Grant or No Patent Grant

Open Source Definition

- For code to be considered open source:
 - The license the code is released under should comply with the properties listed in the definition maintained by the Open Source Initiative (OSI)
- OSI definition
 - 1. Free Redistribution
 - 2. Source Code
 - 3. Derived Works
 - 4. Integrity of The Author's Source Code
 - 5. No Discrimination Against Persons or Groups
 - 6. No Discrimination Against Fields of Endeavor
 - 7. Distribution of License
 - 8. License Must Not Be Specific to a Product
 - 9. License Must Not Restrict Other Software
 - 10. License Must Be Technology-Neutral



OSS Restrictive (Copyleft/Protective) licenses

- Have requirements:
 - How the software can be redistributed
 - Impact how derivative works can be distributed
- The term copyleft originated from the GNU licenses
 - Has been expanded over time to cover other licenses
- Get grouped into strong and weak copyleft licenses

OSS Strong-Copyleft licenses

- Offering people:
 - The right to freely distribute copies and modified versions of a work
 - Provides 4 essential freedoms
 - With the stipulation that the same rights be preserved in derivative works down the line
 - Derivative works, including forks, would need to retain the same license as the original code
 - If you are not the copyright holder of the original code
- Examples:
 - GNU General Public Licenses ([GPL-2.0](#) and [GPL-3.0](#))
 - Affero General Public License ([AGPL-3.0](#))

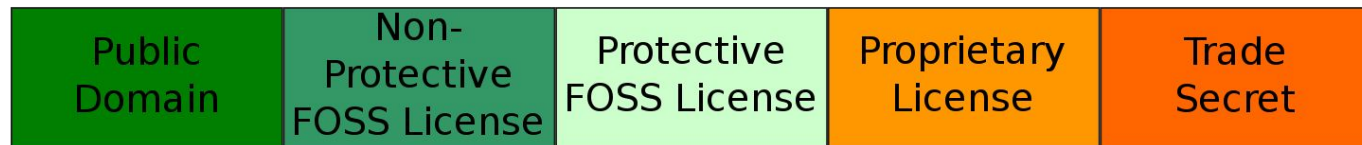
OSS Weak-Copyleft licenses

- Not all derived works inherit the copyleft license
- Generally used for the creation of **software libraries**
 - Allow other software to link to
 - No legal requirement for being distributed under the library's copyleft license
- Examples:
 - Lesser GNU Public License (written as **LGPL-2.0**, **LGPL-2.1** or **LGPL-3.0**)
 - Mozilla Public Licenses (written as **MPL-1.0**, **MPL-1.1** or **MPL-2.0**)
 - Eclipse Public License (written as **EPL-1.0** or **EPL-2.0**)
 - Common Development and Distribution License (written as **CDDL-1.0** or **CDDL-1.1**)

OSS Permissive (Non-protective) licenses

- Minimal requirements on what you must do when redistributing the software
 - Typically limited to things like retaining or delivering attribution notices
- Often easier to be combined with copyleft and/or proprietary software
 - But there still are compatibility issues in some cases
- Common examples:
 - [MIT](#)
 - [BSD-2-Clause](#) and [BSD-3-Clause](#)
 - [Apache-2.0](#)

Software licenses

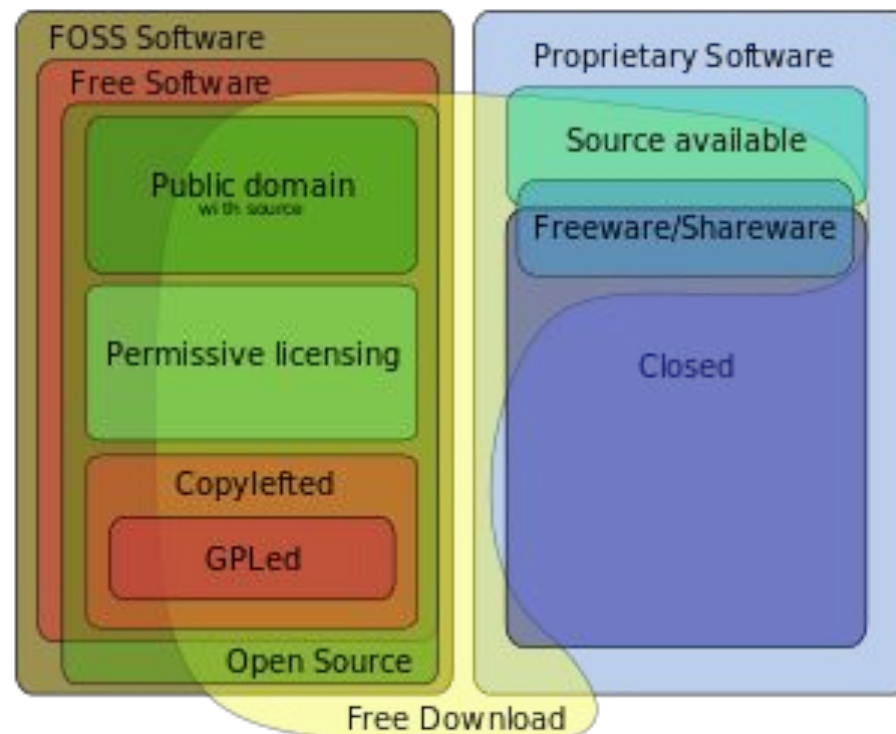


All rights relinquished

← more rights granted

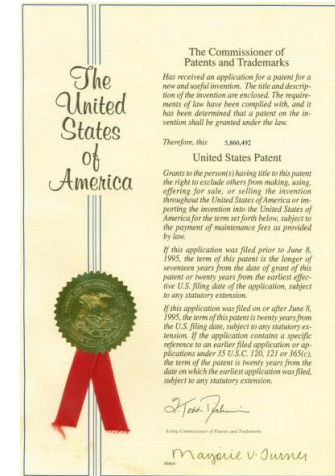
→ more rights retained

All rights retained



Patents

- A mechanism for protecting intellectual property
 - Existed in various forms since the Middle Ages
- Refers to the right granted to anyone
 - Invents any new, useful, and non-obvious process, machine, article of manufacture, or composition of matter
 - Provides the right to exclude others from making, using, selling, offering for sale, or importing the patented invention
 - For the term of the patent
- Must be filed for in each nation (or trading block such as the European Union)
 - Expensive and time-consuming
- [Search for a US patent](#)
- [Google Patents](#)



Patents - Example

- Michael Jackson Smooth Criminal Anti Gravity
 - <https://www.youtube.com/watch?v=2mhnL1YJ2fA>
- Method and means for creating anti-gravity illusion
 - US5255452A
 - Inventor: Michael J. Jackson, Michael L. Bush, Dennis Tompkins
 - <https://patents.google.com/patent/US5255452A/en>

(Software) Patents

- Gives exclusionary rights to material such as a computer program, library, interfaces or even techniques and algorithms
 - Earliest software patents appear to have been granted in early 1960
- Have been used defensively, with corporations cross-licensing each other's work to avoid litigation
 - However, there are many well-known cases of expensive legal battles
 - Existing copyright and trademark laws are insufficient for protecting intellectual property
 - Developers and organizations have to learn to deal with them properly
- Software Patent vs. Copyright
 - Copyright only protects the “exact” source code
 - Avoid infringement by making changes to the code
 - Patents protect the novel features, processes, and design of a system that powers a “software/app” rather than just the underlying source code

Express Patent Grant with License? (1/2)

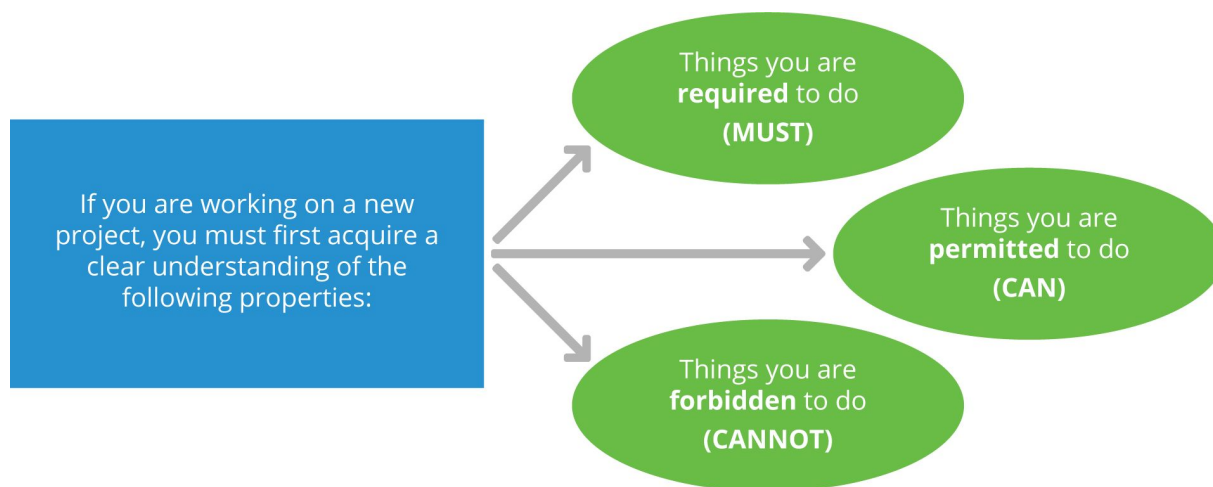
- Software can implement patented inventions
 - Owe the company that patented them a fee to do so
- Some licenses are **EXPLICITLY GRANTED** the right to use any of the patents the contributors have on methods they have included in the code
 - With no additional charge
- Some licenses also **EXPLICITLY EXCLUDE** a patent license grant
- Some licenses that does not explicitly reference patents, but then add a grant of patents in a separate file
 - LICENSE file (BSD-3-Clause)
 - Additional PATENTS file of [Golang repo](#) on GitHub

Express Patent Grant with License? (2/2)

Explicitly EXCLUDES Patent License	Implicit Patent License AND/OR Patents Not Explicitly Mentioned	Explicitly INCLUDES Patent License
<ul style="list-style-type: none">• CC0-1.0• Other Creative Commons licenses, e.g. CC-BY-4.0• BSD-3-Clause-Clear	<ul style="list-style-type: none">• BSD-2-Clause and BSD-3-Clause• MIT• ISC• Apache-1.0 and Apache-1.1• GPL-2.0 and LGPL-2.1	<ul style="list-style-type: none">• Apache-2.0• GPL-3.0 and LGPL-3.0• MPL-1.0, MPL-1.1 and MPL-2.0• EPL-1.0 and EPL-2.0• CDDL-1.0 and CDDL-1.1

Which License to Choose?

- License selected is your way of specifying:



- Choosing a standard and commonly-used open source license
 - Help to make it easier for everyone else to understand what their rights and obligations are

Properties to Consider

- It is important to be clear on your goals for releasing the code
 - Who (what types of people/organizations) do you want to adopt it?
 - Do you want to see any changes people make to your code when they redistribute it?
 - Do you want other people to be able to sell your code for a profit?
- Common properties:

- | | |
|--|---|
| <ul style="list-style-type: none">● Publish license, copyright notices, change summaries?● Disclose source?● Distribution of modified work?● Sublicensing?● Private or commercial use? | <ul style="list-style-type: none">● Patent grant?● Able to use trademarks?● Can code be warrantied?● Able to hold liable for damages?● Scope of license: work as a whole or only specific file? |
|--|---|

Open Source License Comparison

- **Commercial Use:** Software and derivatives may be used for commercial purposes
- **Distribution:** May distribute this software
- **Modification:** May be modified
- **Patent Use:** Provides an express grant of patent rights from the contributor to the recipient. This license explicitly states that it does NOT grant you any rights in the patents of contributors
- **Private Use:** You may use and modify the software without distributing it
- **Disclose Source:** Source code must be made available when distributing the software
- **License and Copyright Notice:** Include a copy of the license and copyright notice with the code

Open Source License Comparison

- **Network Use is Distribution:** Users who interact with the software via network are given the right to receive a copy of the corresponding source code
- **Same License:** Modifications must be released under the same license when distributing the software. In some cases a similar or related license may be used
- **State Changes:** Indicate changes made to the code
- **Hold Liable:** Software is provided without warranty and the software author/license owner cannot be held liable for damages
- **Trademark Use:** This license explicitly states that it does NOT grant you trademark rights, even though licenses without such a statement probably do not grant you any implicit trademark rights

[Open Source License Comparison Grid](#)

Is There Help Available?(1/2)

- [*Open Source Licenses by Category*](#) from the Open Source Initiative lists the approved open source licenses
- [*Choose an Open Source License*](#) is sponsored by GitHub.
 - Walking you through the properties you must consider, helping you decide what license makes sense
 - Blue - Rules you must follow
 - Green - Rules you can follow
 - Red - Things you cannot do
- [*Various Licenses and Comments About Them*](#) provides a description of many licenses and comments about them
- [*Creative Commons Licenses*](#) help you understand license options for images and documentation

Is There Help Available?(2/2)

- The [OSSWatch licence differentiator](#) tool attempts to help its users understand their own preferences in relation to free and open source software licences
- There are 7 choices that need to make
 - 1. Popular and widely used
 - 2. Licence type
 - 3. Jurisdiction
 - 4.a Grants patent rights
 - 4.b Patent retaliation clause
 - 5. Specifies enhanced attribution
 - 6. Addresses ASP/privacy loophole
 - 7. Includes 'no promotion' feature

Mix and Match License?

- When can you combine licenses in a project? There is not a single global answer to this question
 - Interpretations of license compatibility
 - Which licenses are acceptable for inclusion, depend on the project and distribution the project ships in
 - Not all projects treat combinations of licenses the same way
- Guidance created by some of the common projects
 - [GNU/Free Software Foundation](#)
 - [Fedora Project](#)
 - [Debian](#)
 - [Android Open Source Project](#)
 - [The Apache Software Foundation](#)

REMEMBER:
You should always check with your legal counsel before contributing to a project on behalf of your employer

What Does a Reference to a License Look Like in a File?

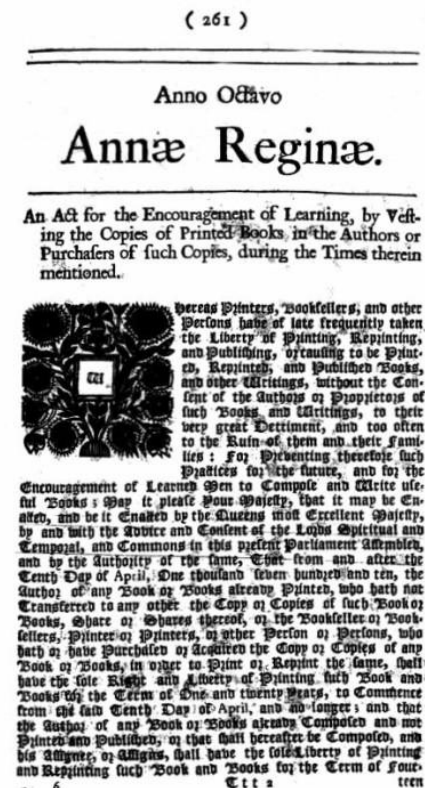
- Full license text is frequently long and complex,
 - NOT always suitable to put in each file in a project
- Common practice:
 - Put a LICENSE file at the top level of the project,
 - Include the full text of the license(s) that may be in effect for the project as a whole
 - For the other files in a project,
 - Referring to the actual license within each file — even if that reference doesn't include the full license text
 - Ambiguity is removed as to which license governs that file,
 - Particularly when there are multiple licenses being used by a project
 - A license reference actually look like in a file
 - A standard header for the license (if it exists)
 - An unambiguous reference to the license intended

What Is a Copyright? (1/2)

- "Copyright is a *legal right* created by the law of a country that grants *the creator* of an original work *exclusive rights for its use and distribution*."
(according to Wikipedia)
 - Granting to the copyright owner for protection of their work
 - Original "work" can refer to creative, intellectual or artistic output.
 - Software source code
 - Copyright does not cover the ideas or information, but rather how they are expressed (or written down)
 - Ideas and algorithms can be covered by patent law instead
 - The work must be considered as original
- Countries have their own copyright laws
- International treaties (e.g. Berne Convention and the Universal Copyright Convention) have standardized the treatment
 - Local variations in law do exist
 - United States did not sign the Berne Convention until 1989
 - Over 100 years after the Berne Convention was created (1886)

What Is a Copyright? (2/2)

- Great Britain was the first country to recognize the rights of authors
 - The Copyright Act of 1709, or Statute of Anne, was the first law that recognized the rights of authors
 - The law has been reworked and refined over time as technologies changed
- Copyright protection does have limits. In the United States, it does not extend to:
 - Any idea, procedure, process, system, principle or discovery
 - Facts such as names, titles, short phrases, slogans, familiar symbols, typographic variations, lettering, coloring and listing of contents or ingredients



Who Is a Copyright Holder?

- The author of the work is generally considered to be the copyright holder
- When a work was created in the course of the author's employment?
 - It is considered as a “work for hire” and the employer is the copyright holder
- Rights of the copyright holder can also be assigned to another organization or individual
- Some jurisdictions around the world may have different positions on who owns a copyright,
 - Should always check with local counsel first

Copyright Notices

- Under the Berne Convention, an author doesn't need to register their copyright in order to own it
 - The copyright exists upon the creation of the work
- Still a good idea to insert a copyright notice in the body of the work
 - Make it clear that the work is copyrighted
 - Provide guidance about who the copyright holders are

Historical Format:

Copyright (C) 2001, 2004-2006 Company ABC.

Copyright 1998, Linus Torvalds

© 2003, 2010, Free Software Foundation, Inc.

Copyright (C) 2011, 2014-2019 Company ABC.

Newer Format

Copyright Contributors to the _____ Project.

Copyright The _____ Contributors.

Copyright The Kubernetes Authors.

Copyright Contributors to the OpenVDB Project.

Copyright the Hyperledger Fabric contributors.

What Are Valid File Notices?

- Valid file notices have the following characteristics:
 - Define in a clear manner the license that the file is governed by
 - May include information or references to copyright holders
 - Also provide additional information about authorship, when the author is not the copyright holder
- Examples:
 - [Linux](#)
 - [FOSSology](#)
 - [gcc](#)
 - [Nova](#)
 - [Das U-Boot](#)
 - [Zephyr](#)