



# Open Source Software Engineering

**SE ZG587**

**Session 4**

**BITS Pilani**  
Pilani Campus

Kumar Manish  
12<sup>th</sup> August, 2023



**BITS Pilani**  
Pilani Campus



# Understanding Software Licenses in OSS

# Software Licenses



- A software License is a **Legal instrument** which describes the manner in which a software can be used or redistributed - in both source code or object code forms.
- A typical software license grants the **Licensee** ( end-user) , the permission to the software in a manner where such a use would **otherwise potentially constitute copyright Infringement of the software** owner's exclusive rights.
- **Software Licensing Models :**
  - Copyrights
  - Copyleft Licenses - Protective
  - Permissive Licenses - More free
  - Creative Commons Licenses - Public copyrights licenses

# Copyrights



- Copyright is an important intellectual property license that gives the owner **an exclusive right on the work created by him / her**. The owner is allowed to create copies of the creative work
- The creative work may be of any type - including literary, artistic, educational, or musical form
- As a general rule, for works created after January 1, 1978, copyright protection lasts for **the life of the author plus a additional 70 years**.
- In case of a **corporate authors**, the protection is for the shorter of **95 years from publication or 120 years from creation**

- Copyleft is a licensing method,
  - used to make a work ( or program ) free to use, modify , adapt or extend
  - The freedom to carry out all activities , as aligned with the **four essential freedoms** ( prerequisite – source code available)
  - Copy left licenses includes the GNU general public license ( GPL) – which was originally written by Richard Stallman
  - the work or program should be **made available to the recipients** - **Mostly in the form of source code**
- No re-licensing allowed, No commercial usage allowed
- All modified and extended version of the work ) or program ) should be free as well - hence called **protective licenses**
- Rules for copyleft –
  - all derivative works should be attributed to the creator, open sourced and copyleft
- Copyleft is a generic concept , and can't be used directly ; one needs to use a specific implementations of the concepts

# Variants of Copyleft

- **Strong and Weak copyleft** : The strength of the copyleft license is decided based on the extent its provision are imposed on the derived works

## Examples :

- Strong Copyleft :
  - GNU general public License
- Weak copyleft
  - GNU Lesser General Public License –
  - Mozilla public license

## Weak copyleft licensing :

- Most commonly, weak copyleft licenses are used to create **Software Libraries**
- Help software to link to the library and redistributed without requirement for the linking software to also be copyleft –licensed

# Copyleft – share alike condition



## Share-alike :

- Imposes the requirement that any freedom that is granted regarding the original work must be granted on exactly the same or compatible terms in all derived work
- However, in some cases the author may be willing to share only a certain part of the work, **but the share – alike agreement require that the whole body of the work is shared**
- The positive side – for an author of source code – any modification to the code **will not only benefit the original creator, but that the author will be attributed and recognised and hold equal claim over the changed code.**

# Permissive Licenses



- Free software licenses with only minimal restrictions on how the software can be used , modified and redistributed ( also called Berkeley software distribution : BSD-like or style licenses )
- Rules for usage - what ever user wants , but with few restrictions – derived works must be attributed to the creator
- Source code need not be open or made available in the public domain
- Re-licensing allowed - derivative works can be release under any other licenses or used as proprietary products
- Allows commercial usage
- Examples :
  - GNU All permissive License
  - MIT License
  - BSD licenses
  - Apple Public source licenses
  - Apache license



# Comparing Permissive and Copyleft

Copyleft (Protective Licenses )	Permissive Licenses
Publication of software code of all modified versions or derived works under the <b>original copyleft licenses</b> - protective licenses	Provides No guarantee that derived works of the software will remain free and publicly available ; generally requiring only that the original copyrights notice be remained
	<ul style="list-style-type: none"> <li>- Hence derived works , or future versions , of permissively – licenses software can be released as proprietary software</li> <li>- Permissive licenses offer highly extensive license compatibility as compared to copyleft licenses</li> <li>- Wider adaptability in the open source community</li> </ul>



# Creative Common Licenses ( CCL)

- A public copyright licenses that **enable free distribution of copyright work** – allowing the users the right to share, use , and build upon a work that – someone else ( the author ) has created
- **Rules for usage** - whatever user wants without any restrictions - **derived works must be attributed to the creator**
- Availability of the source code : No specific terms about the distribution of source code
- **Re-licensing allowed**
- **Commercial usage allowed** , however author can decide to allow uses of given work as non commercial
- CCL applied all works : books , plays, movies, photograph , music, articles, blogs and websites
- CCL were initially released in 2002 by creative commons , Non profit Org founded in 2001
- So far Five version of licenses , latest in 2013 - version 4

# Creative Commons Licenses



## Compatibility with software :

- **Not recommended for use for software** - since they **do not contain specific terms about the distributions of source code**, which is important in order to ensure the free reuse and modifiability of software
- **Most of the CC licenses are not compatible with the major software licenses** , so it would be difficult to integrate **CC licenses work with other free software**
- However, they may be used for software documentations , or other artistic elements embedded within documentation.
- **Free Software Foundations (FSF)** : In 2011 added “CC 0 version”
- **Open Source Initiative (OSI)** : Not approved under OSI because of its clause **which excluded from the scope of the licenses any relevant patents held by the copyrights holder.**



# How to choose an open source License ?

---

- Open source licenses help in protecting works contributed in the open source domain
- It also help protect contributors and users from any copyright infringements
- Interested developers or contributors or business will not touch a project without a license protection.

Which license should I choose :

- If I wish to work with a community
- If I wish to keep the licenses simple and permissive
- If I wish to share the improvements made
- If I wish to work without license

# Recap



- Intellectual Property Rights and Software Licenses
- Licensing Models in OSS:
  - Copyright,
  - Copyleft,
  - Permissive,
  - Creative Commons



# Understanding and Choosing Open Source Licensing Models

# Choosing Open Source Licensing Models

---

- Option 1: Work with a community
- Option 2: Keep it simple and permissive
- Option 3: Need to share improvements
- Option 4: Work without a license



# How to choose an open source License ?

---

## Why to choose ?

- Open source licenses help in protecting works contributed in the open source domain
- It also help protect contributors and users from any copyright infringements
- Interested developers or contributors or business will not touch a project without a license protection.

## Which License should I choose :

- If I wish to work with a community
- If I wish to keep the license simple and permissive
- If I wish to share the improvements made
- If I wish to work without a license





# Work with a community

- To contribute to or modify or adapt an existing projects,
  - Continue contributions using original license available with the project
- Moreover, using the same license might also be a requirement
  - as per stated in the original license terms of the project
- Look at LICENSE or COPYING or README file for more info
- Certain Open source software projects require contributors to sign the **Contributors License Agreement (CLA)**
  - Simple CLAs vs Detailed CLAs
  - Individual CLAs vs Corporate CLAs
- No centralised body of knowledge engaged in standardizations of CLAs
  - Different organisations write their own versions
  - Larger project need formal CLAs
  - Generally baked by one or more corporations

Ex – Apache, Django, Eclipse Foundations etc

# Work with a community



- Normally, a CLA require the contributors to make certain classifications ( In terms of legal proofs) , which may comprise of one or more of the following :
  - the contributors is the author of the contributions;
  - the contribution is an original work;
  - the contribution is not subject to third – party license , claims etc;
  - the contribution has the legal right to grant the copyright license;
  - the contributors does not have an employer that can claim rights in the copyright;
- Benefits of CLA : protection against copyright infringement
- Disadvantages : discourage contributions

# Work with a community



## How to manage the ownership and licensing of copyrights for individual software contributors to the project ?

**Option 1 :** The author ( who is also, by default, the copyright holder) of the software contributions retains the software 's copyright and ownership and contributes it under the same open source license as used by the project.

**Option 2 :** The contributors assigns the ownership of the software copyright to The “**Project Maintainer**”. Who then releases the software under the projects' Open Source License.

**Option 3 :** the last option is not to define any ownership policy

# Keep it simple and permissive



- Choose the MIT License :
  - It is a short and simple permissive license with condition only requiring preservation of copyright and license notices.
- Licensed works, modification and larger works may be distributed under different terms and without source code.
- The MIT license allows users the permission to reuse code for any purpose , or embed it as a part of proprietary software or make any changes or modifications to the code to suit their needs.
- Restrictions - users are required to include the original copy of the MIT licenses

# Keep it simple and permissive



- **Permissions :**
  - Commercial Use : the licenses materials and derivatives may be used for commercial purposes
  - Distribution : The licensed materials may be distributed
  - Modification : The licensed materials may be modified
  - Private use : The licensed materials may be used and modified in private
- **Conditions :**
  - A copy of the license and copyright notice **MUST** be included with the license materials.
- **Limitations :**
  - Liability : The license includes a limitation of liability
  - Warranty : The license explicitly states that it does **NOT** provide any warranty

# Need to share improvements



- Choose the GNU general Public Licenses v3.0 – copyleft license - protective license
- This strong copyleft license provisions on making available complete source code of licensed works and modifications, including larger works under the same license.
- Copyright and license notice Must be preserved.
- Contributors provides an express grant of patent rights.
- GPL assures that patent can not be used to render the program non free.

# Need to share improvements

- **Permissions :**

- Commercial use : The licensed materials and derivatives may be used for commercial purposes.
- Distributions : the licensed materials may be distributed.
- Modification : The licensed materials may be modified
- Private use: The license materials may be used and modified in private
- Patent use : the license provides an express grants of patent rights from contributors

- **Limitation :**

- Liability : The license includes a limitation of liability
- Warranty : The license explicitly states that it does NOT provide any warranty

# Need to share improvements



## Conditions :

- Disclose source : Source code **MUST** be made available when the licensed materials is distributed.
- License and copyright notice : A copy of the license and copyright notice **Must** be include with the license material.
- Same license : Modifications **Must** be released under the same license when distributing the licensed materials . In some cases a similar or released license may be used.
- State Changes : Changes made to the licensed materials **Must** be documented



# Work without a license

- By default, all works are under exclusive copyrights of the creator – be it creative work or code
- Unless a license is associated with the copyrighted works, that states otherwise, no one can copy, distribute, or modify the work without being at risk of copyright infringement
- However, in the absence of a license file, one may still grant some rights – this holds in cases where one publishes the source code on a site that requires accepting terms of service
- For example, if the source code is published in a public repository on GitHub, and you have accepted their terms of service, by this you allow others to view and fork your repository

# Work without a license

- You do not have to do anything to work without a license
  - However, in the case , one may wish to add a copyright notice and statement - indicating that no license has been included - in a prominent place – e.g. the project README
- Moreover, disallowing the use of your code might not be what you actually intent by “ no license”
  - An open source license allows reuse of your code while retaining copyright.
  - Add a [contributors license agreement \(CLA\)](#) to your non-licensed project, so that you maintain copyright permission from contributors, even though you are not granting the same.

# Work without a license



To use or contribute to project that has no license associated with it

- Do not use the software : Find or create an alternative that is under an open source license
- Request the Project Maintainers to add a license :
  - Unless the software included strong indications to the contrary, lack of a license is probably an oversight.
    - If the software is hosted on a site like GitHub, open an issue requesting a license
    - If you are sure what license is most appropriate, open a pull request to add a license - with a suggestive license

# References and further readings

---

- Choose a license <https://choosealicense.com/>
- Open Source Initiative <https://opensource.org/>
- Open Source Resources <https://opensource.com/>
- Open Source Guides (<https://opensource.guide/>
- Creative commons <https://creativecommons.org>
- GNU <https://www.gnu.org/>
- Copyleft <https://copyleft.org/>