



Software License and free software

Alexandre Bergel

<http://bergel.eu>

16/12/2020

Relevance of software licenses

I am not a lawyer, and I do not read licenses...

So, why should I care about licenses?

A software license is a legal instrument governing the use of redistribution of software

The software license indicates how to use work from others

A license allows for other to use a particular work

Intellectual property

Often refer to various privileges awarded over *intangible* goods with economic value

Include concepts such as *copyright* to protect from unauthorized copy literary or artistic works, computer programs, data compilations, industrial designs.

Trademarks protect symbols

Geographical indication protect appellations of origin

Trade secret protects the hiding of information

Patent to enable temporary monopolies to inventions in exchange for their revelation

Intellectual property

Article 27 of the Declaration of Human Rights acknowledges

“everyone has the right freely to participate in the cultural life of the community, to enjoy the arts and to share in scientific advancement and its benefits”

It further acknowledges the right for protection of the model and material interests result from production

http://www.unesco.org/culture/culture-sector-knowledge-management-tools/11_Info%20Sheet_Cultural%20Rights.pdf

Software license

A software license is *legal instrument* governing *the use or redistribution* of software

All of a software is *copyright protected*, in both *source code* and *compiled / executable code*

Choosing a license

Choosing a license is a *very important decision* and requires careful thought

Switching to a different license later during the life of the project *could be difficult, if not impossible*

E.g., many contributors with rights

Public-domain software

A software with *no ownership*, *trademark*, or *patent*

Software in the public domain can be modified, distributed, or sold even *without any attribution by anyone*

Creative Commons Public Domain Mark



Software freedom

Anyone commercializing a program imposes the conditions under which the program can be used

Software is the most flexible and adaptable item of technology we have

It is therefore possible to impose prohibition to adapt it to particular needs, or to correct its error, without the explicit agreement of the manufacturer

Free software offers freedoms that *proprietary software* denies

Definition of Free Software

Free software, as conceived by Richard Stallman

<http://www.gnu.org/philosophy/free-sw.html>

Refers to four freedoms granted to its receivers:

- 1- freedom to run the program in any place, for any purpose and forever
- 2- freedom to study how it works and to adapt it to our needs. This requires access to the source code
- 3- freedom to redistribute copies, so that we can help our friends and neighbors
- 4- freedom to improve the program and to release improvements to the public. This also requires the source code

Freedoms guarantee

In accordance with current legislation, the way to *guarantee these freedoms* is to distribute the software under a specific license

Through the license, the author gives permission for the receiver of the program to exercise these freedoms, adding also any restrictions that the author may wish to apply

E.g., to credit the original authors in the case of a redistribution

Ambiguity of the term free

The English term *free software* includes the word *free*, standing for *freedom*

But *free* also mean *free of charge*

Which is why in some case, the English borrow Spanish/ French words and refer to *libre software*, as opposed to *gratis software*

The definition of free software make no reference to the fact that it may be obtained free of charge: free software and gratis software are two very different things.

Motivations for defining free software

Ethical motivation

Argue that software is knowledge that should be shared without obstruction. Being able to modify program is a form of freedom of expression.

Pragmatic motivation

Championed by the open source initiative. Argues many technical and financial advantages

Adopting a free license

May have many other reasons:

Fun (e.g., Linus Torvalds)

Money with sustainable business models

Consequence of the freedom of software

Very different *business model* than proprietary software

Quality resulting from the *voluntary collaboration* of contributing people

Discover and fixing bugs

Collaboration and competition combine in order to produce better quality

Financial consequence

It is not possible to make much money from distributing a free software

Which means that money is made by someone other than the authors

This is why particular business models are financing mechanism are needed

Free software cost model is very different to the proprietary software cost model

Two types of free licenses

Permissive licenses: do not impose special conditions on the second redistribution. Software can be modified and allows someone to redistribute it as a proprietary software

Strong licenses (also called copyleft license): impose conditions when redistributing the software. Redistribution must comply with the license condition of the first redistribution.

Permissive licence

Guarantee of *maximum freedom for the person receiving the program*

Maximum neglect in respect or ensuring that the person offers the same freedoms when redistributing that program

Allow software that its author distributes under a permissive licence to be redistributed with a proprietary license.

Many permissive licenses exists: XWindow license, version 11 (X11), Zope Public License 2.0, Apache license, the BSD Licence, *the MIT license*

The MIT License

Copyright <YEAR> <COPYRIGHT HOLDER>

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.

The 3-Clause BSD License

Copyright <YEAR> <COPYRIGHT HOLDER>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Difference between BSD and MIT

Practical differences between the 2-clause BSD license and the MIT license are marginal.

What both the 2-clause BSD license and the MIT license have in common are:

- Permits use

- Permits redistribution

- Permits redistribution with modification

- Provision to retain the copyright notice and warranty disclaimer

In addition the MIT license also explicitly allows:

- merging

- publishing

- sublicensing

- selling

Strong licenses

General Public license of the GNU project (GPL) is the most popular and well-known in the world of free software

Instead of limiting users rights, GPL guarantees them

The GPL license allows redistribution in binary form *and* in source code

It is only possible to redistribute GPL code under other code that has a license compatible with GPL

You cannot include GPL code in a proprietary software

This is called the *viral effect* (although many consider it as disrespectful)

But?

OSX (and many other OS) does include GNU GPL code and without providing source code

Is Apple against the law?

The answer is no

The GNU General Public License, version 2 says:

"For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally."

<https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html>

But?

Apple was trying to get rid of GPL-Licensed packages on some point...

<https://news.ycombinator.com/item?id=3559990>

Money and software distribution

Due to the third freedom, *anyone can redistribute a program without asking for a financial reward* or permission

Which makes it practically impossible to obtain big profits just by distributing free software

Economy

Free software is developed in many different ways and the way to obtain funds varies enormously from case to case

Every free project has its own way of financing itself:

- From volunteer developers

- From altruistically funds

- From public funding

- Private not-for-profit funding

- Financing by someone requiring improvements

- Funding with related benefits (books, hardware)

How to pay your rent?

“How to pay your rent with your open source project”

<https://plausible.io/blog/open-source-funding>

Creative Commons

<https://creativecommons.org> is a very popular framework to build permissive licenses

Applicable to text, book, lecture notes, documents, music, pictures

Not advised to use it for software, because it does not contain specific terms about the distribution of source code

The WTFPL license

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
Version 2, December 2004

Copyright (C) 2004 Sam Hocevar <sam@hocevar.net>

Everyone is permitted to copy and distribute verbatim or modified copies of this license document, and changing it is allowed as long as the name is changed.

DO WHAT THE FUCK YOU WANT TO PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. You just DO WHAT THE FUCK YOU WANT TO.

Climate Strike License

Copyright (c) <year> <copyright holders>

"Climate Strike" License Version 1.0 (Draft)

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 limitation in 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 may not be used in applications and services that are used for or aid in the exploration, extraction, refinement, processing, or transportation of fossil fuels.

The Software may not be used by companies that rely on fossil fuel extraction as their primary means of revenue. This includes but is not limited to the companies listed at <https://climatestrike.software/blocklist>

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.

<https://choosealicense.com>

Choose an open source license

An open source license protects contributors and users. Businesses and savvy developers won't touch a project without this protection.

{ Which of the following best describes your situation? }



I need to work in a community.

Use the [license preferred by the community](#) you're contributing to or depending on. Your project will fit right in.

If you have a dependency that doesn't have a license, ask its maintainers to [add a license](#).



I want it simple and permissive.

The [MIT License](#) is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

[Babel](#), [.NET Core](#), and [Rails](#) use the MIT License.



I care about sharing improvements.

The [GNU GPLv3](#) also lets people do almost anything they want with your project, *except* distributing closed source versions.

[Ansible](#), [Bash](#), and [GIMP](#) use the GNU GPLv3.

{ What if none of these work for me? }

My project isn't software.

[There are licenses for that.](#)

I want more choices.

[More licenses are available.](#)

I don't want to choose a license.

[Here's what happens if you don't.](#)



Lerna

A tool for managing JavaScript projects with multiple packages.



Star

25,859

[Follow Lerna on Twitter](#)

The case of the Lerna project

In 2018, one of the Lerna project code maintainers made some changes in the license definition

The maintained made a pull request that changed the used MIT license to include *restrictive* and *retroactive* clauses (<https://github.com/lerna/lerna/pull/1616>)

Not only personal uses were impacted, but also tech companies, including Microsoft, amazon, Dell, Ceros, Canon, LinkedIn and few other were not allowed to use Lerna

The case of the Lerna project

The motivation for the maintainer is these tech companies were collaborating with the US Immigration and Customs Enforcement (“ICE”)

The maintainer said: *“these companies care only about the millions of dollars that ICE is paying them and are willing to ignore all the horrible things that ICE does”*

The case of the Lerna project

1 - restricting the use of the software to certain groups of users, Lerna is discriminating. Non-discrimination is a core value of open source.

Fifth clause of the Open Source Definition says: “license must not discriminate against any person or group of persons”

2 - there is no such thing as a retroactive license change. The Free Software definition is ambiguous when it says that:
“if the developer of the software has the power to revoke the license, or retroactively add restrictions to its terms, without your doing anything wrong to give cause, the software is not free”

The case of the Lerna project

That single pull-request had several repercussions

“The Lerna project’s choice is, moreover, destructive of one of the deep norms that keep the open-source community functional - keeping politics separated from our work” — Eric Raymond

Open source developers, when not coding, are free to do whatever they want

It does not matter if the developers supports ICE or a far right-wing party

Mixing politics with open source resulted in a major impact

The case of the Lerna project

The pull request *was reverted*

The maintainer was *kicked out* of the project

The open source community around Lerna *was hurt*

Why? By using Lerna to advocate on his own behalf, the maintainer also indicated that the Lerna project shared his political view

This contribute *negatively to attract new developers* that may not share the same political view.

=> All this for just 22 lines added in the license file



OPEN SOURCE LICENSING 101

FROM CONCEPTS TO
PRACTICAL IMPLICATIONS

GUSTAVO PINTO

Closing words

Software licensing is an important topic that should not be overlooked. Many companies and open source communities are very careful.

The MIT license is the most popular license.

MIT license is compatible with many business models. With a MIT-licensed product:

You CAN (i) re-use the code freely for your own use, (ii) re-use the code for non-commercial AND commercial re-distribution, whether in source OR binary form

You CANNOT: (i) claim authorship of the software, (ii) thus you cannot attack the original author for using or publishing his original version

Closing words

MIT is basically a simple contract that says:

Person or company X created Y

Y belongs to X, but X is granting you the right to use it and do whatever you want with it

X cannot be held accountable for anything that goes downhill with what you do with Y

License



Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

You are free to:

- Share: copy and redistribute the material in any medium or format
- Adapt: remix, transform, and build upon the material for any purpose, even commercially

The licensor cannot revoke these freedoms as long as you follow the license terms



Attribution: you must give appropriate credit



ShareAlike: if you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original

Complete license: <https://creativecommons.org/licenses/by-sa/4.0/>