

**Open Source Licenses**

Open Source Software for Science, Business and Management

Course Project

Done by: EKSfmu-16 gr. st. Arūnas Butkus

Checked by: doc. dr. Raimond Laptik

Contents

[1. Introduction 3](#_Toc498161594)

[2. GNU licenses 3](#_Toc498161595)

[3. MIT license 3](#_Toc498161596)

[4. Other licenses 3](#_Toc498161597)

[5. No license 3](#_Toc498161598)

[6. Conclusions 3](#_Toc498161599)

# 1. Introduction

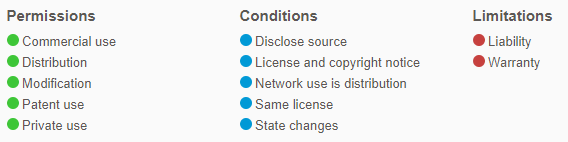
Make something official sounding and overviewing the content in things paragraph.

# 2. GNU licenses

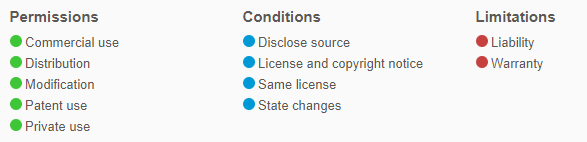
There are few types and versions of GNU licenses. From most restricting Affero General Public License (AGPL), to General Public License (GPL), to least restricting Lesser General Public License (LGPL). GNU licenses are also versioned now because the first GNU license was released on 1989 February 25th, originally written by Richard Stallman of the Free Software Foundation for the GNU Project, and software, legal, licensing, etc. changed a lot since then so significant updates to the original license were required. Currently used is version 3 published on June 29th, 2007 – after roughly 17 months of deliberations.

GNU Licenses are widely used free software licenses, which guarantee end users the freedom to run, study, share and modify the software. The licenses grant the recipients of a computer program the rights of the Free Software Definition – computer software distributed under these terms allow users to run the software for any purpose as well as to study, change, and distribute it and any adapted versions. GNU GPL was the first copyleft license for general use and being copyleft means that derivative work can only be distributed under the same license terms [ref gnu wiki].

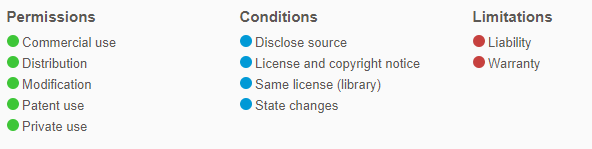
As far as permissions and implementation goes, AGPL is most limiting. Permissions of this license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Also in cases when modified version is used to provide a service over a network the complete source code of the modified version must be made available. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.



Base GPL license is slightly less strict. Permissions of this copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.



LGPL is most lenient of GPL licenses. Permissions of this license are conditioned on making available complete source code of licensed works and modifications under the same license or the GNU GPLv3. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work through interfaces provided by the licensed work may be distributed under different terms and without source code for the larger work. Meaning this one can be part of a commercial work that does not disclose its source code.

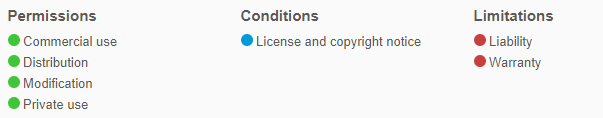


Thing with GPL licenses is that they are copyleft – derivative, or just work that uses software licensed under GPL in most cases also has to have GPL license. This is problematic when working on commercial projects as usually you don’t really want to give away the source code of a program that you’re trying to sell. That is why it has fallen out of favor quite a bit.

# 3. MIT license

Lately I see it is really popular to use MIT license. And Black Duck research shows quite comprehensively that MIT is getting ahead in popularity. From sample set of over 2 million open source forges and repositories 32% software has MIT license while only 18% utilize GPLv2; though adding all the variants of GPL licenses it adds up to almost 32% as well. Though considering the rate of change in use MIT license GPL will be completely eclipsed soon enough, since in 2014 GPLv2 alone was quite significantly more popular than MIT and by 2015 MIT caught up in popularity with GPLv2. [ref zed webpage thing].

MIT is a short, simple copyright license with few restrictions that only require preserving the copyright and license notices themselves within used source codes. Code using this license can easily be used in larger, licensed works or modifications under different terms and without providing the source code itself. MIT license is also compatible with many copyleft licenses, such as the aforementioned GPL – MIT licensed software can be integrated into GPL software, but not the other way around.

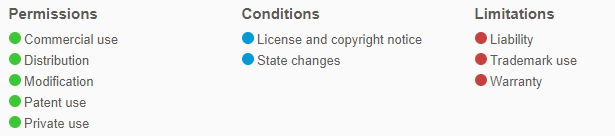


Name MIT originates from the Massachusetts Institute of Technology (MIT) – institute that released the license. Notable projects that use some variant of the MIT License include Ruby on Rails, Node.js, jQuery, and the X Window System.

# 4. Other popular software licenses

There are also a few more licenses that are starting to make GPL obsolete. According to Black Duck data 14% of software use Apache License 2.0 and 6% use 3-Clause BSD License. There is also a push for Mozilla Public License 2.0 (MPL) for example.

Apache License 2.0 released in January of 2004 is another permissive copyright license like MIT. Main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without providing original source code.



BSD licenses are a part of copyright licenses group of permissive free software licenses, imposing minimal restrictions on the use and redistribution of covered software. The original BSD license was used for its namesake, the Berkeley Software Distribution (BSD), a Unix-like operating system. But the original version has been revised a few times since then and its descendants are more properly termed modified BSD licenses, though still retaining large parts of the original. BSD is both a license and a class of license (generally referred to as BSD-like). The BSD license is a simple license that merely requires that all code licensed under the BSD license be licensed under the BSD license if redistributed in source code format. BSD (unlike some other licenses) does not require that source code be distributed at all.

# 5. Variety of open source licenses

List and explain a few from the list: https://opensource.org/licenses/category

# 6. Conclusions

Write some proper conclusion before opinion.

Personally, only reason I see use of open source license is if, for whatever reason, a company would release its licensed software source code as open source, to ensure the work not entirely ripped off. But honestly you either not disclose source code when selling the program or just put up your snippet of code unlicensed with easy access for anyone – if someone finds it useful and rips it off, good for him. Then again, might as well drop MIT license text block somewhere in there to make your code look more official/professional.

https://choosealicense.com/licenses/

https://www.blackducksoftware.com/top-open-source-licenses

https://en.wikipedia.org/wiki/GNU\_General\_Public\_License#Version\_1

http://www.zdnet.com/article/the-fall-of-gpl-and-the-rise-of-permissive-open-source-licenses/