

Open Source Licenses

Disclaimer:

The information in this work has been prepared by Hama Othman for informational purposes only, is not legal advice and should not be relied on as legal advice. The information in this work does not necessarily reflect the views of Cordemeyer & Slager / Advocaten B.V. or its clients. There is no guarantee that the information is accurate, complete or up-to-date. Laws and procedures change frequently and are subject to differing interpretations. This work is not intended to substitute for obtaining legal advice from competent, independent, legal counsel in the relevant jurisdiction. If you want legal advice, please consult Cordemeyer & Slager / Advocaten B.V. personally.

Author: Hama Othman

February 2017

Index

1. INTRODUCTION	p. 4
2. WHAT IS FREE/OPEN SOURCE SOFTWARE	p. 5
2.1 What is software?	p. 5
2.2 Difference Between Open and Closed Source Software	p. 6
2.3 Difference between Free and Open Source Software	p. 7
2.3.1 Free Software Foundation	p. 8
2.3.2 Open Source Initiative	p. 8
3. WHAT IS AN OPEN SOURCE LICENSE?	p. 11
3.1 Intellectual Property	p. 11
3.2 Common Open Source Licenses	p. 12
3.2.1 BSD licenses	p. 12
3.2.2 MIT License	p. 13
3.2.3 Apache 2.0 License	p. 13
3.3.3 GPL	p. 13
4. DIFFERENCES BETWEEN OPEN SOURCE LICENSES	p. 15
4.1 Permissive v. (weak) Copyleft	p. 15
4.2 Rights and Conditions Under the Different Licenses	p. 20
4.2.1 MIT License	p. 20
4.2.2 BSD licenses	p. 22
4.2.3 Apache 2.0 License	p. 22
4.2.4 GPL	p. 24
4.2.4.1 GPLv2	p. 25
4.2.4.2 GPLv3	p. 27
4.2.4.3 AGPLv3	p. 29
4.2.4.4 LGPLv2.1 and 3	p. 29

4.3 Compatibility of Licenses	p. 31
4.2.1 Compatibility of Open Source Licenses	p. 31
4.3.2 Open Source Licenses and Proprietary Licenses	p. 32
 5. LEGAL APPROACH TO OPEN SOURCE LICENSES	 p. 33
5.1 Applicable Law and Jurisdiction	p. 33
5.2 Enforcement of Open Source License	p. 35
 6. CASE LAW OPEN SOURCE LICENSES	 p. 37
6.1 The Netherlands	p. 37
6.1.1 Curry / Audax (Amsterdam, 2006)	p. 37
6.2 Elsewhere in the European Union	p. 38
6.2.1 Welte / Sitecom (Munich, 2004)	p. 38
6.2.2 Welte as a third party in AVM / Cybits (Berlin, 2011)	p. 39
6.2.3 Welte / Fantec (Hamburg, 2013)	p. 39
6.2.4 Hellwig / VMware Global Inc. (Hamburg, 2016)	p. 40
6.3 United States	p. 42
6.3.1 Jacobsen v. Katzer (9th Cir., 2006-2010)	p. 42
6.3.1 Busybox cases (S.D.N.Y., 2007-2012)	p. 43
 7. HOW TO COMPLY WITH OPEN SOURCE LICENSES?	 p. 45
7.1 MIT License	p. 46
7.2 BSD 2- & 3-clause License	p. 46
7.3 Apache 2.0 License	p. 47
7.4 GPL	p. 47
7.4.1 GPLv2	p. 47
7.4.2 GPLv3	p. 48
 Appendix A – Rights, Conditions and Limitations	 p. 49
Appendix B – Compatibility of Open Source Licenses	p. 50

1. Introduction

Open source software is everywhere. It's in the Cloud¹, it's in our phones², and maybe even in our cars³. The most-used mobile operating system in the world, android, is licensed under both the GPLv2 and the Apache 2.0 License. But what exactly does open source mean, and what are the differences between the open source licenses? These are the type of questions that will be answered in the remainder of this work.

This work is meant to introduce you to the world of open source software and, more specifically, to the world of open source licenses surrounding it. In Chapter 2, the basic questions – what software means, what open source software means and what the differences are between closed and open source software – will be answered. Chapter 3 will explain what an open source license is, and what the current most popular open source licenses are. In Chapter 4, the differences between many of the open source licenses will be examined and the question of what compatibility of licenses entails will be answered. Chapter 5 will deal with the legal aspects surrounding open source licenses. In Chapter 6, case law concerning open source licenses will be examined and finally, Chapter 7 will provide a summary of this work in a visual form.

¹ Content management platforms which power most of the websites we use, such as WordPress, Drupal and Joomla, are all licensed under different open source licenses.

² The whole android operating system is licensed under open source licenses. Apple's iOS operating system, while not open source, is making use of many open source components nevertheless.

³ See: <<https://media.ford.com/content/fordmedia/fna/us/en/news/2017/01/04/ford-and-toyota-establish-smartdevicelink-consortium.html>> (accessed February 18, 2017).

2. What is Free/Open Source Software?

2.1 What is Software?

It is crucial to have an understanding of what software is before one can dive into the complex universe of software licenses. Essentially, software is everything that is executed through hardware. Examples of software include: a computer operating system such as Microsoft Windows (closed source software), its corresponding free and open source counterpart Linux, a program such as Internet Explorer (closed source software), and its corresponding free and open source counterpart Mozilla Firefox.⁴

Software is written in programming languages such as Java, Python, and PHP.⁵ These programming languages make it possible for software developers to write source code. Source code is human-readable and looks similar to a normal language.⁶ Once the programmer has finished writing their program in the programming language, they then will save the source code as text. To make the program run, what needs to happen next is translation of the source code into machine-readable code. This is done using programs called *compilers*. A compiler translates everything that the programmer has written in the human-readable programming language into a machine-readable-only language.⁷ The result of this “translation” is that the programmer will be left with computer-readable “object code.” It is this object code that makes programs run on our computers, smartphones, smartwatches etc. However, once source code is translated into object-code, it can no longer be modified. This means that it is impossible for users of the program to change or improve the program. Therefore, one must have access to the source code to be able to do those things.⁸

⁴ Vasudeva, V. (2014). Open source software and intellectual property rights. 1st ed. Alphen aan den Rijn: Wolters Kluwer Law & Business, p. 12.

⁵ There are many more programming languages, but these are the three most popular ones according to GitHub's *Popularity of Programming Language*, <http://pypl.github.io/PYPL.html>, accessed 23 January 2017.

⁶ Meeker, H. (2015). *Open source for business: a practical guide to open source software licensing*. 1st ed. North Charleston, SC: CreateSpace Independent Publishing Platform, p. 12.

⁷ Ibid., p. 12-13.

⁸ Ibid., p. 14.

2.2 Difference Between Open and Closed Source Software

As noted in the previous paragraph, software is written in human-readable programming languages (source code) and is then compiled into a machine-readable code (object code). The main difference between open source software and closed source software lies in the fact that users, generally,⁹ do not have access to the source code for closed source software (or proprietary software) whereas users do have access to open source software source code. To illustrate this difference, one can look at the classic example of Microsoft's closed source operating system, Windows, and its corresponding open source counterpart Linux.

In article 2(c) of Microsoft's Windows Operating System License Terms,¹⁰ several restrictions, to which users are deemed to have agreed when using the software, are specified. The restrictions include, but are not limited to: making copies of the software is prohibited, reverse-engineering is prohibited, and using the software for certain purposes, (for example, as server software) is prohibited. Subsequently, anyone who uses Microsoft Windows does not have access to the software's source code. In fact, it is against the license terms to (try to) decompile Microsoft Windows' object code.

Microsoft Windows is a perfect example of closed source software. On the other hand, Linux is a perfect example of open source software. Just like Microsoft Windows, Linux too is an operating system but, unlike Microsoft Windows, Linux uses an open source license for the software viz., GPLv2.¹¹ While the GPL licenses will be covered in greater detail in the following chapters, it is necessary to illustrate the significant differences between Linux's license terms and Microsoft Windows' license terms.

In article 1 and 2 of the GPLv2¹² license, the following rights are explicitly granted to the software user: the right to copy, modify and distribute either changed or unchanged versions

⁹ Usually only the developers of the software, e.g. developers at Microsoft working on a program, have access to the source code of said program.

¹⁰ The entire license agreement can be accessed through: <https://www.microsoft.com/en-us/Useterms/Retail/Windows/10/UseTerms_Retail_Windows_10_English.htm> (accessed February 18, 2017).

¹¹ The Linux Kernel Archives. (n.d.). Frequently asked questions. Retrieved February 18, 2017, from <<https://www.kernel.org/category/faq.html>>.

¹² GNU Operating System. (n.d.). GNU General Public License, version 2. Retrieved February 18, 2017, from <<https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>>.

of the program. However, to enjoy the rights granted by the GPLv2 license, a few conditions must be complied with. These conditions are: **disclosure of the source code when distributing the software**, a license and copyright notice, **the same license must be used when distributing the software in either changed or unchanged form**, and changes to the software must be stated. The two conditions noted in bold make the GPLv2 license a so-called “copyleft” license, as opposed to the other category of open source licenses, the so-called “permissive licenses” (copyleft and permissive licenses will be discussed in the following chapters). Therefore, as long as a user complies with the conditions listed under a GPLv2 license, they have the right to copy, modify and distribute the software as they see fit. On the other hand, as soon as a user violates one of the terms, the rights granted under GPLv2 will be automatically terminated (the consequence of this will be covered in the following chapters).¹³

In conclusion, closed source software has licenses that restrict the use of the software and only allow for distribution, modification and/or reproduction of the software in special cases. Open source software has open source licenses that grant the user of the software most¹⁴ of the rights that copyright holders enjoy, as long as they abide by the license terms.

¹³ Article 4 GPLv2.

¹⁴ Apart from the exclusive right to assign copyright, open source license users enjoy all the rights which usually only the original author of copyrighted material enjoys e.g. reproduction rights, modification rights, distribution rights, and performance rights.

2.3 Difference Between Free and Open Source Software

In the world of open source licensing, two organizations have played a pivotal role in shaping the landscape. These two organizations are the Free Software Foundation (FSF) and the Open Source Initiative (OSI).

2.3.1 Free Software Foundation

In the early days of computers and software, the sharing of software source code was the normal course of business. However, in the early 1980s this sharing economy came to a halt when big corporations discovered that big money could be made with software programs. Companies such as IBM, Microsoft, and AT&T started making software with proprietary licenses attached to them. In most of these cases, this meant that users of software programs had to sign a non-disclosure agreement stating that they would not share any of the source code with other people.¹⁵

Richard M. Stallman (RMS) was working at MIT as a programmer during this transitional period. RMS disapproved of the way things were going (i.e. with users of software being required to sign non-disclosure agreements to be able to use the software) and decided that he wanted to do something about the situation. He started developing an operating system that would be free of any of the conditions set out in a proprietary license. This operating system would be named GNU and was designed to be very similar to the then-popular operating system, UNIX.¹⁶

To prevent GNU software from ever being licensed under a proprietary license, RMS came up with the idea of “copyleft”. Copyleft, a play on the word copyright, “is a method of making a program free, and requiring all modified and extended versions of the program to be free as well.”¹⁷ The first license to implement this method was the GPLv1 license written by RMS in 1989. After GPLv1 was released, a new updated version of the license was released

¹⁵ M Stallman, R. (n.d.). The GNU Project. Retrieved February 18, 2017, from <<https://www.gnu.org/gnu/thegnuproject.en.html>>.

¹⁶ Ibid.

¹⁷ GNU Operating System. (n.d.). What is Copyleft? Retrieved February 18, 2017, from <<https://www.gnu.org/licenses/copyleft.en.html>>.

in 1991 under the name GPLv2. This license became the de-facto license for open source projects licensed under a copyleft license.

Ever since the development of GNU, the FSF has been a fierce advocate for free software. However, “free” in the sense of “free software” does not refer to price. It refers to freedom – the freedom to do whatever you want with the software. More precisely, it refers to four freedoms that a user of the program must have:

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this, you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

A program is free software if all of these freedoms are adequately provided to users. Otherwise, it is classified as “non-free”. While we can distinguish various non-free distribution schemes in terms of how far they fall short of being free, the FSF considers them all equally unethical.¹⁸ It is important to note that “free software” has mostly been associated with the GPL copyleft licenses.

2.3.1 Open Source Initiative

The Open Source Initiative (OSI) is the second organization that has played a pivotal role in the landscape of open source licensing. The OSI was created in 1998 in response to the FSF. OSI’s main goal was to make open source more palatable to the corporate community instead of catering to “geeks” and hackers only.

¹⁸ GNU Operating System. (n.d.). What is free software? Retrieved February 18, 2017, from <<https://www.gnu.org/philosophy/free-sw.en.html>>.

The main difference between the OSI and the FSF is their software philosophy. The FSF believes that copyright is bad and that all software should be free.¹⁹ On the other hand, the OSI has a more pragmatic view of software. The OSI does not believe that intellectual property rights are bad, and does not believe that all software should be open source. The OSI has, to a large extent, succeeded in making open source software popular not just for hackers, but also for big technology companies around the world.²⁰

The OSI is now the de-facto organization that assesses whether a software license is an open source license or not. If a license fulfils ten requirements found on the OSI's website, then the license can call itself an open source license.²¹ These ten requirements are very similar in spirit to the FSF's four freedoms. However, unlike the FSF, the OSI does not place a significant emphasis on "copyleft".

In the remainder of this work, the terms "free software" and "open source software" are used interchangeably.

¹⁹ M Stallman, R. (n.d.). <Why Software Should Not Have Owners. Retrieved February 18, 2017, from <https://www.gnu.org/philosophy/free-sw.en.html>>.

²⁰ Companies such as Google, Facebook, Microsoft and Apple are huge supporters of different open source projects.

²¹ Open Source Initiative. (2007, March 22). The Open Source Definition. Retrieved February 18, 2017, from <<https://opensource.org/osd>>.

3. What is an Open Source License?

3.1 Intellectual Property

To understand what open source licenses are and what they try to achieve, it is important to have an understanding of intellectual property rights. “Intellectual property rights” is an umbrella term that encompasses many legal rights. However, the essential intellectual property rights that can apply to software are rights under: trade secret law, patent law, trademark law and copyright law.²² The most important intellectual property rights for open source licenses are rights under copyright law.

Although copyright law is specific to every country, the general principles are more or less the same around the world through the Berne Convention.²³ The Berne Convention is a treaty that is applicable to most countries in the world (for example, US, EU, India, China). The most important and relevant rule of the Berne Convention for open source licensing is that copyright automatically arises in a work as soon as the work is expressed (i.e., it is on paper, on film, on disk, or on a website). Furthermore, registration is not required for copyright to arise.²⁴ Therefore, as soon as someone writes source code and saves it on a computer, puts it online or even writes it in a notebook, copyright automatically arises on the written code. Based on the Berne Convention, the minimum exclusive rights that every author of copyrightable works enjoys include “the right to make adaptations, arrangements and other alterations to the work”²⁵ and “the right to make reproductions in any manner or form”.²⁶

Because copyright arises automatically, licenses are needed to grant users basic rights so they can make use of the software without infringing copyright. Therefore, licenses are used to grant users rights that they normally don’t have. On the other hand, licenses are also being used to create conditions for users of the software.

²² Vasudeva, V. (2014). Open source software and intellectual property rights. 1st ed. Alphen aan den Rijn: Wolters Kluwer Law & Business, p. 31.

²³ Long title: Berne Convention for the Protection of Literary and Artistic Works.

²⁴ Article 2(2) Berne Convention.

²⁵ Article 12 Berne Convention.

²⁶ Article 9 Berne Convention.

3.2 Common Open Source Licenses

Currently, there are 80 open source licenses approved by the Open Source Initiative.²⁷ However, most of these licenses are not popular or widely used nor do they have strong communities. According to data from GitHub²⁸, more than 80 percent of the projects hosted on its website use either the MIT License, one of the GPL licenses, the Apache 2.0 License or one of the BSD licenses.²⁹ GitHub data is also consistent with the data analyzed by Black Duck³⁰. According to Black Duck, almost 75 percent of the open source projects either use the MIT license, GPLv2 license, Apache 2.0 license, GPLv3 license, or the BSD licenses.³¹ Therefore, the most common open source licenses currently in circulation, are:

3.2.1 BSD licenses

The BSD licenses are a family of open source software licenses imposing minimal restrictions on the redistribution of BSD license-covered software. The original BSD license was released by the University of California, Berkley, to permit the free use, modification, and distribution of certain University of California software without any return condition whatsoever on the part of licensees.³² The most popular software project licensed under the BSD license is the Berkley Software Distribution, an operating system developed and distributed by a research group from the University of California, Berkley.

The original BSD License has been revised several times, and thus, many versions of the license are in circulation. The most popular versions, however, are the 3-clause BSD License, and the 2-clause BSD License. The BSD licenses are simple licenses with, as the names suggest, two or three clauses. The licenses allow users to copy, modify and distribute BSD licensed work and have very easy conditions to comply with. About five percent of the open source projects make use of the BSD licenses.

²⁷ Open Source Initiative. (n.d.). Licenses by Name. Retrieved February 18, 2017, from <<https://opensource.org/licenses/alphabetical>>.

²⁸ GitHub is the most popular website for hosting open source projects.

²⁹ Balter, B. (2015, March 09). Open source license usage on GitHub.com [Blog post]. Retrieved February 18, 2017, from <<https://github.com/blog/1964-open-source-license-usage-on-github-com>>.

³⁰ Black Duck is a software company that specializes in open source license compliance and governance.

³¹ This data reflects analysis by Black Duck of over two million open source projects from over 9,000 global forges and repositories, <<https://www.blackducksoftware.com/top-open-source-licenses>> (retrieved February 18, 2017).

³² Rosen, L. (2005). Open source licensing. 1st ed. Upper Saddle River, NJ: Prentice Hall PTR, p. 73.

3.2.2 MIT License

The MIT License was written by lawyers at the Massachusetts Institute of Technology (MIT) and is based on the BSD license.³³ It is functionally very similar to the BSD license; however, the MIT license is much simpler to read and understand. The whole license consists of a mere 171 words and grants users the rights to copy, modify and distribute MIT licensed work, with very easy conditions to comply with. The MIT license is the most used open source license with 30–45 percent of open source projects using it.

3.2.3 Apache 2.0 License

The Apache 2.0 License is similar in function to the BSD licenses and the MIT license. It also is a license that grants users the rights to copy, modify and distribute Apache 2.0 Licensed work and has easy conditions to comply with. However, the Apache 2.0 License looks much more like a legal document when compared to the BSD licenses and the MIT License. The Apache 2.0 License consists of nine sections, with the first section listing the definitions of terms used in the license. The Apache 2.0 License is a very popular license because of its comprehensiveness. It is used by about 10–15 percent of open source projects. Android, Google’s popular and widely-used mobile operating system is, for the most part, licensed under Apache 2.0.³⁴

3.2.4 GPL

The GPL (officially the GNU General Public License) is perhaps the best known and most feared open source license by businesses and lawyers. The first version of the GPL was written by Richard Stallman in 1989 and was used for the GNU Project.³⁵ The second version of the GPL was written in 1991 and the third version, which is the current version, was written in 2007. The most used versions of the GPL are Version 2 and Version 3, with Version 2 being the more popular version. However, the GPL license includes an optional “any later version” clause, allowing users to choose between the original terms or the terms in new versions of a GPL license. Because of this optional clause, many open source projects can also be licensed under GPLv3, even though they were initially released under GPLv2.

³³ Ibid., p. 85.

³⁴ Android Open Source Project. (n.d.). Licenses. Retrieved February 18, 2017, from <<https://source.android.com/source/licenses.html>>.

³⁵ The GNU Project is a free software collaboration project started by Richard M. Stallman at MIT.

The GPL license is a so-called “copyleft” license. This means that derivative works based on GPLv2/3 licensed software can only be distributed under the same license terms. In simpler terms, you are free to use GPLv2/3 code; however, if you use this code, you must make your own code free also. This is in contrast to so-called “permissive licenses” such as the BSD licenses, MIT License, and Apache 2.0 license, which are the most frequently used permissive open source licenses.

The GPL license was the first general use copyleft license and is currently the most widely-used copyleft license in circulation. Apart from the GPL license, variants based on the GPL exist viz., the Lesser/Library GPL (or LGPL), and the Affero GPL (or AGPL). These two variants are important additions of the GPL license family. About 25–30 percent of open source software projects make use of one of the variants of the GPL license family, with the operating system Linux³⁶ being the best-known example of GPL licensed software.

³⁶ Linux is licensed under GPLv2.

4. Differences Between Open Source Licenses

4.1 Permissive v. (weak) Copyleft

As outlined above, there is a distinction between permissive open source licenses and copyleft licenses. Copyleft licenses, however, can further be divided into weak copyleft licenses, strong copyleft licenses, and ultra-strong copyleft licenses.

Permissive licenses are very simple licenses. In essence, under a permissive license you are allowed to do whatever you want with the licensed software as long as you abide by the notice requirements. Like every other open source license, permissive licenses provide the software without any warranties and contain a limitation of liability. Therefore, permissive licenses can be summarized as follows:³⁷

- Do whatever you want with the code;
- Use the code at your own risk;
- Acknowledge whoever wrote that piece of software.³⁸

The most popular and most used permissive licenses are the MIT license, BSD licenses, and the Apache 2.0 license. There are many more permissive licenses, and licenses such as the MIT License have hundreds of other permissive licenses based on it. However, because they are all permissive licenses, they generally all work the same way.³⁹ Because its conditions are easy to comply with, permissive licenses are generally compatible with both open source licenses and proprietary licenses. In other words, you can take source code licensed under a permissive license and use the code for your closed source project. Usually, when developers opt for permissive open source licenses it is because they want their code to be used by as many people, and in as many ways, as possible.

Copyleft licenses, on the other hand, are a bit more complicated but, in essence, copyleft licenses have the same conditions as noted above and include:

³⁷ Meeker, H. (2015). *Open source for business: a practical guide to open source software licensing*. 1st ed. North Charleston, SC: CreateSpace Independent Publishing Platform, p. 31.

³⁸ Ibid.

³⁹ Ibid.

- If you distribute binaries (software in object code), you must make the source code for those binaries available;
- The source code must be available under the same copyleft terms under which you got the code;
- You cannot place additional restrictions on the exercise of the license.⁴⁰

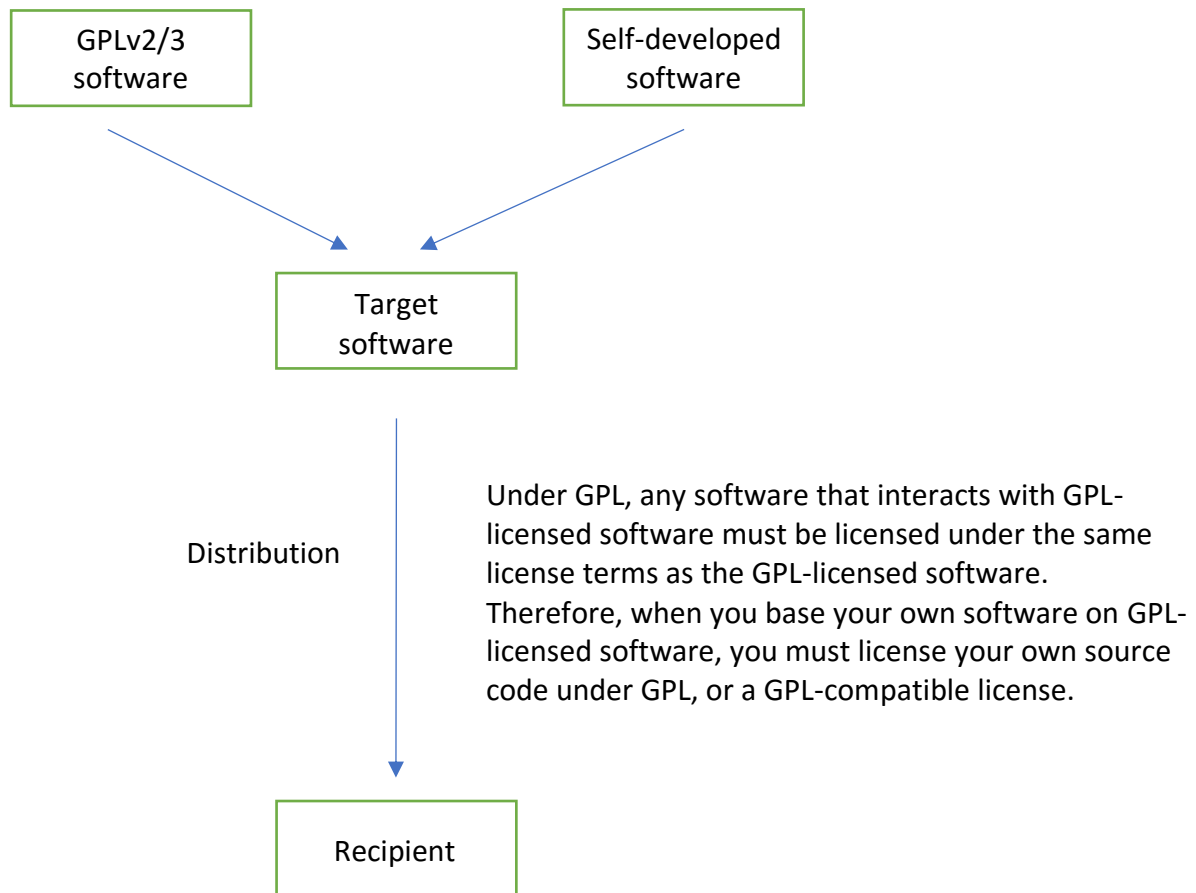
Copyleft licenses can be divided into different categories ranging from ultra-strong copyleft licenses to weak copyleft licenses. The AGPLv3 license is considered an ultra-strong license. GPLv2/3 are considered strong copyleft licenses. LGPLv2/3 are considered weaker copyleft licenses, and uncommon licenses such as the Eclipse, MPL and CDDL licenses are considered weak copyleft licenses.⁴¹

The strength of the copyleft refers to the scope of the software that is subject to the copyleft license terms. In other words, a strong copyleft license such as GPLv2/3 requires that any software program that contains GPL licensed code be subject to the GPL. This is sometimes also referred to as the “viral” nature of GPL. LGPL, on the other hand, allows dynamic linking to LGPL licensed software. If this is the case, the LGPL license only applies to the LGPL-licensed software project, while the other software project that links to the LGPL licensed software project can be licensed under any license. Weak copyleft licenses such as the Eclipse license or MPL license don’t have the viral characteristic that the GPL licenses have. The AGPLv3 license has the same scope as the GPLv2/3 license. However, as will be explained further in the next paragraph, you must only comply with the GPL license conditions when distribution occurs. Under GPLv2/3, SaaS distribution is not considered distribution, whereas under AGPL, SaaS is considered distribution and therefore the AGPL license conditions also apply in those cases.

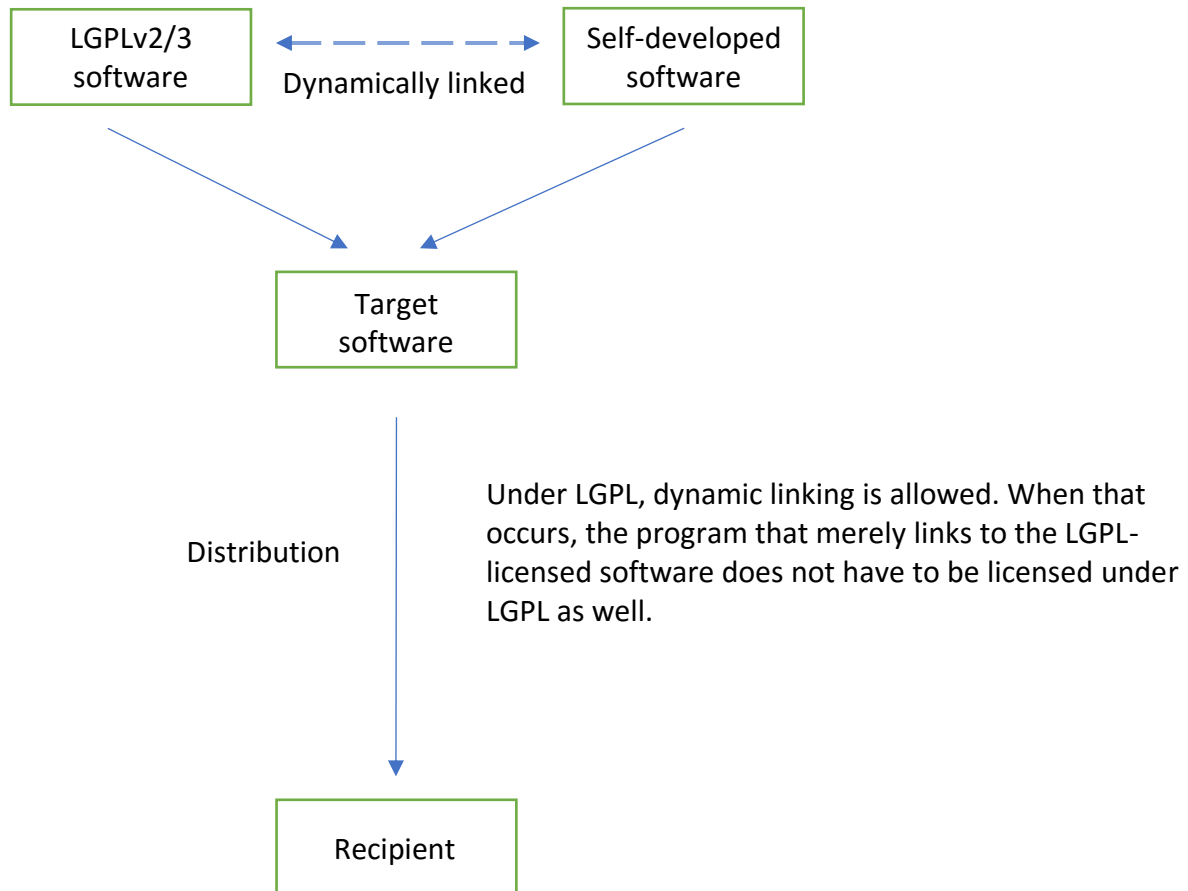
⁴⁰ Ibid.

⁴¹ Ibid.

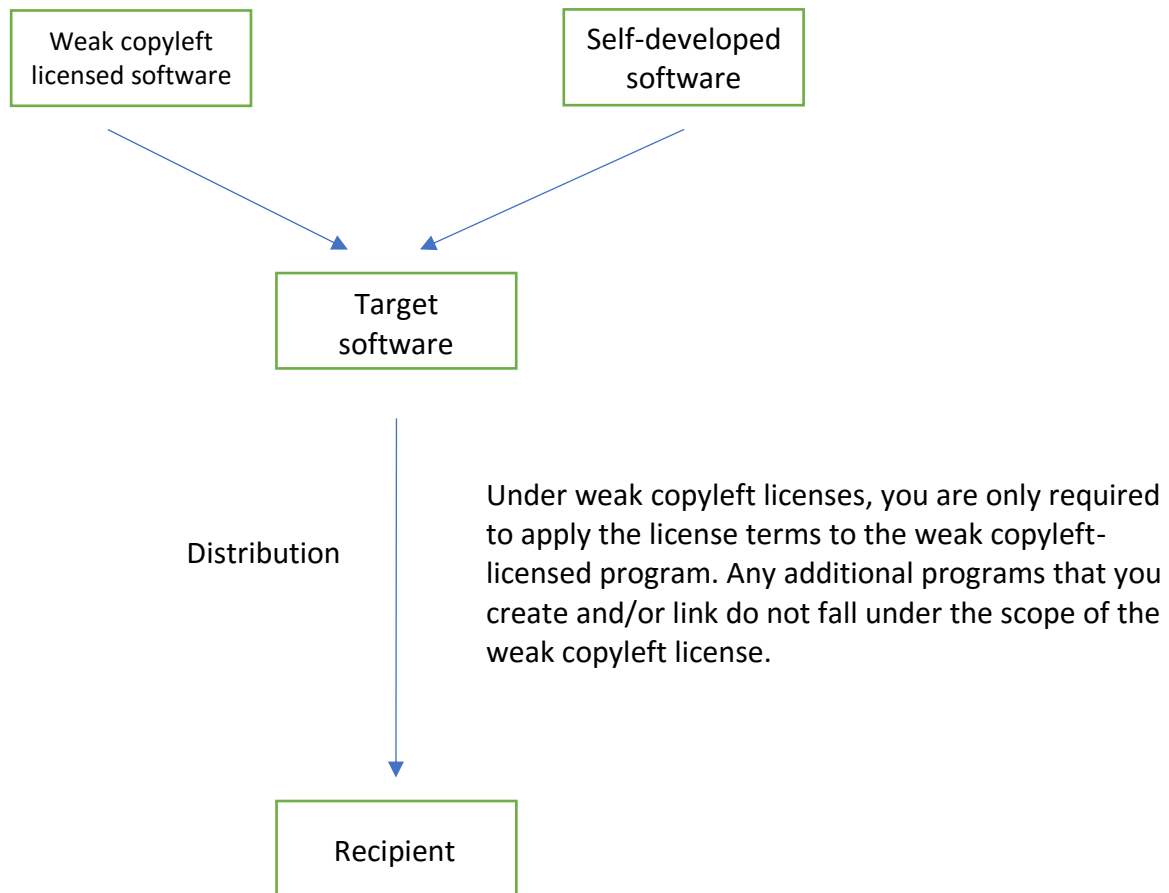
The following diagram illustrates what strong copyleft entails:



As explained above, the LGPL license allows dynamic linking. The following diagram illustrates what that entails:



The following diagram illustrates what weak copyleft entails:



4.2 Rights and Conditions Under the Different Licenses

4.2.1 MIT License

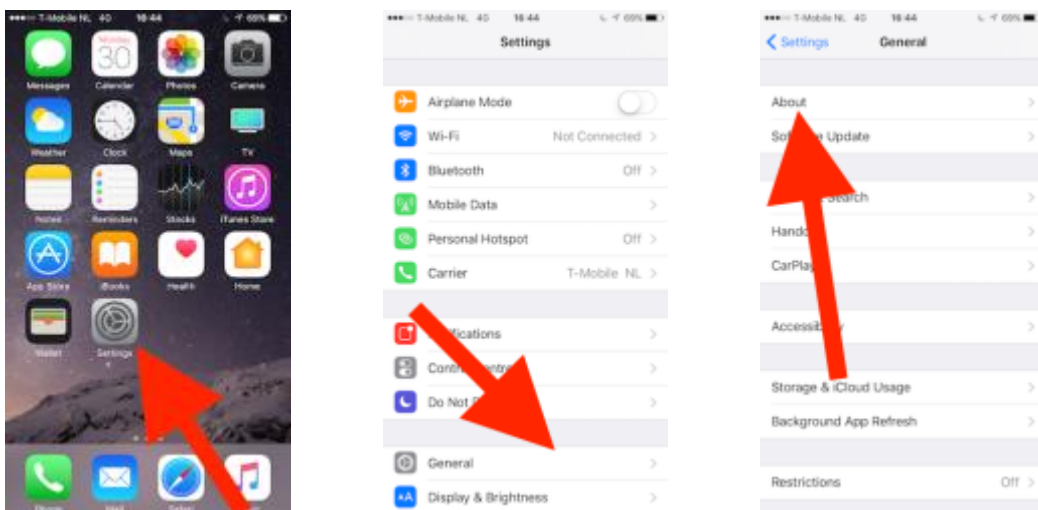
As noted previously, the MIT License is one of the simplest licenses to read and understand. It is a license with no individual sections, but instead comprises three short paragraphs. In the first paragraph, the users of the license are granted, free of charge, the following rights without restrictions: to *use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software*.

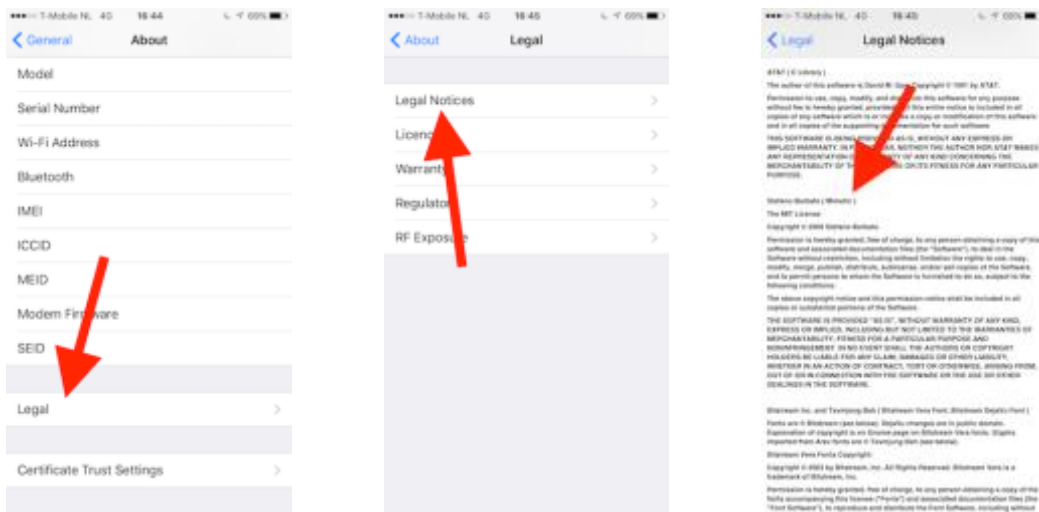
The only condition under the MIT License is:

“The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.” “The above copyright notice” refers to this piece of text:

“Copyright <YEAR> <COPYRIGHT HOLDER>” that must be filled in by the author of the source code.

Developers or businesses can easily comply with this condition. To illustrate this, we can look at how a company such as Apple has complied with the condition found in the MIT License. Apple’s iPhone operating system, iOS, makes use of many open source licensed software programs. The following pictures illustrate how Apple has dealt with its notice requirement conditions.





As can be seen, Apple is using, among many others, a piece of open source software called Mimetic, which is developed by Stefano Barbato, and is licensed under the MIT License in its mobile operating system. Apple is complying with its license conditions by bundling all the copyright and license notices in one place called “Legal Notices.” In reality, very few users will look at those notices, in part because they are quite hidden away. However, by making it possible for users to see the notices, Apple is complying with its notice obligations under the open source licenses. This particular way of hiding the notices is how almost every major company complies with its license conditions when incorporating software licensed under permissive licenses in their products.

Apart from the notice requirements, the MIT license does not have any other conditions. However, apart from the grant of rights and the notice requirements, the MIT license also contains a paragraph detailing the limitation of warranty. In short, this means that when you are using the MIT-licensed source code, neither the original author of the source code, nor any later contributors can be held liable for anything that goes wrong with the source code.

A developer, seller, manufacturer etc. can contractually agree upon different terms. For example, it is possible that a seller of a product that contains MIT-licensed software agrees that he will be liable if the product and/or its software does not work properly. This obligation, however, is purely contractual and is only binding on the contracting parties, in this case the seller and the buyer.

4.2.2 BSD licenses

The BSD licenses are also very popular licenses in the open source world. The first BSD license was created in 1988 and since then many alternations have been made to the original license. Therefore, when people speak of a BSD License, they either refer to the BSD 3-clause license (or BSD 3-clause “New” or “Revised” License) or the BSD 2-clause license. The BSD 2-clause and 3-clause licenses are very similar to the MIT License. In fact, the BSD 2-clause License grants the same rights and has the same conditions as the MIT License, but is organized and worded differently. The BSD 3-clause License is, for the most part, the same as the BSD 2-clause License although it has a third clause, which states:

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.

Therefore, under BSD 3-clause licensed source code, you can do everything you are allowed to do under either an MIT License or a BSD 2-clause License, but the 3-clause license prohibits you from using the name of the copyright holder or its contributors to promote derived products without written consent from them. Furthermore, both the 2-clause and 3-clause licenses oblige you to include a copy of the license and copyright notice when distributing the source code.

As with any other open source license, the BSD licenses contain a paragraph limiting the liability of the author and contributors of the source code.

4.2.3 Apache 2.0 License

The Apache 2.0 License is, just like the MIT License and BSD licenses, a permissive open source license. However, unlike the other permissive licenses, the Apache 2.0 license spans three pages. The reason for this is because the Apache 2.0 License looks a lot like a true legal document, with sections, subsections etc. The whole license is split into different sections, with the first section being a definitions section. In Section 2 of the license, the following rights are granted to users of Apache 2.0 licensed work:

to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

These are basically the same rights that all the other open source licenses grant users. However, the Apache 2.0 License has quite an interesting, and important, clause that most of the other permissive licenses don't. This particular clause can be found in Section 3 of the License, titled "Grant of Patent License". Section 3 is quite a long clause, but the most important passages are the following:

[...] each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable patent license [...] to use the Work, [...] applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) [...]. If You institute patent litigation against any entity [...] alleging [...] patent infringement, then any patent licenses granted to You [...] shall terminate [...].

This clause makes it more difficult for so-called "patent trolls" to take advantage of the open source community. Contributors to an Apache 2.0-licensed project not only contribute code, they also explicitly grant users of that code a license to use any of their patents that would otherwise be infringed upon by the code that was contributed. Furthermore, if one of the patent owners sues someone over patents related to the Apache 2.0-licensed software, then the owner who sues will immediately lose any other patent licenses that might have been granted to them by other contributors of the Apache 2.0-licensed project.

In addition to the explicit patent grant clause, the Apache 2.0 is unique in another way. Indeed, it is one of the few permissive licenses that explicitly prohibits the use of any trademark rights that might be present in the open source project licensed under the Apache 2.0 License. This rule is laid down in Section 6 of the license. However, even though many other licenses don't have this explicit prohibition on trademark use, such licenses probably don't grant a user any implicit trademark rights either.

Section 4 of the License lays down the conditions under which a user is allowed to distribute an Apache 2.0-licensed project. The first condition is to provide recipients of the Apache 2.0 licensed software with a copy of the Apache 2.0 License.⁴² The next condition is that whoever significantly modifies (parts of) the Apache 2.0-licensed project is required to state those modifications.⁴³ Furthermore, when distributing the Apache 2.0-licensed work, any copyright, patent, trademark and/or attribution notices must be retained.⁴⁴

As can be seen, the Apache 2.0 License does not have the requirement to license any modified versions of the Apache 2.0-licensed project under the same license. It is therefore possible to use Apache 2.0-licensed code in a closed source project without having to make your own code available to the recipients of the software. However, you are still required to comply with the abovementioned conditions, even if you license your own version of the project under a proprietary license.

Finally, as with every other open source license, Apache 2.0-licensed works are provided “as is”, meaning that there are no warranties. Furthermore, all liability is disclaimed insofar as it is possible under the applicable local law.⁴⁵ It is, however, possible to offer additional support in the form of warranty, indemnity and/or any other liability conditions. However, in doing so, it is required to “indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.”⁴⁶ Essentially, this means that you are required to agree to indemnify any other contributors of the work if you offer additional support to a recipient of an Apache 2.0-licensed work.

4.2.4 GPL

The GPL licenses are the most used copyleft licenses currently adopted by the open source community. Currently, the most used GPL license is the GPLv2 license. However, in addition to the GPLv2 license, there is also the successor to the GPLv2 license, the GPLv3 license. These two licenses are considered strong copyleft licenses. As a consequence, the FSF has also created the LGPL license, which is considered a weak copyleft license. There are two

⁴² Section 4(1) Apache 2.0 License.

⁴³ Section 4(2) Apache 2.0 License.

⁴⁴ Sections 4(3), 4(4) Apache 2.0 License.

⁴⁵ Sections 7, 8 Apache 2.0 License.

⁴⁶ Section 9 Apache 2.0 License.

variants of this LGPL license: the LGPLv2.1 and the LGPLv3. Then there is the AGPLv3 license, which is considered an ultra-strong copyleft license due to the way “distribution” is interpreted in that license. These licenses will be discussed below, starting with the GPLv2 license.

4.2.4.1 GPLv2

The GPL licenses also grant users of GPL-licensed software the right to copy, distribute, and modify the work. However, under the GPLv2 license you are required to share the changes that you have made to the code under the same conditions of the licensed work.⁴⁷ Essentially, this means that you must disclose the source code of your modifications of the GPLv2-licensed software when you distribute (e.g. sell) the software. This means that, unlike permissively licensed code, which people can take and use in their closed-source software to sell, GPL-licensed software spreads to all the code that is attached to the GPLv2-licensed software (derivative works). This property is often described as the “viral” nature of the GPL licenses.

However, one key factor to keep in mind is that the virality of the GPL licenses only kicks in when you distribute the GPL-licensed software. It is therefore important to understand what is meant by “distribution” under the GPL licenses. Unfortunately, the GPLv2 license does not provide a definition of the term “distribution”. The authors of the license assumed courts would adopt the definition of “distribute” from US copyright law.⁴⁸

Under US law, SaaS use does not constitute distribution, as under US law an actual transfer of a copy is required. Accessing software without transfer of a local copy to the user is therefore not considered distribution.⁴⁹ Every act in which a third party gets a local copy of the actual software, however, is considered distribution.

When distribution occurs, the following conditions apply:

⁴⁷ Section 3 in conjunction with section 4 GPLv2.

⁴⁸ Smith, B. (n.d.). A Quick Guide to GPLv3. Retrieved February 18, 2017, from <<https://www.gnu.org/licenses/quick-guide-gplv3.html>>. (“The license never says what distribution is, because the term was borrowed from United States copyright law. We expected that judges would look there for the definition.”)

⁴⁹ Meeker, H. (2015). *Open source for business: a practical guide to open source software licensing*. 1st ed. North Charleston, SC: CreateSpace Independent Publishing Platform, p. 72.

1. Display of copyright notice, disclaimer of warranty, and requirement to give any recipients of the program a copy of the GPLv2 license.⁵⁰
2. If you have changed the program, you must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.⁵¹
3. If you modify the GPLv2-licensed software and your modification becomes a “derivative work”, then you are obliged to license the derivative work under the same license conditions as the original program.⁵²
4. You must provide the source code of the software program to the recipient of the software.⁵³
5. You are not allowed to change the license or introduce additional terms and conditions.⁵⁴

The most important question regarding the GPL licenses, therefore, is: what is considered a derivative work? Businesses and organizations are afraid to use GPL-licensed software because of the “viral” risk (i.e. when a company is using GPL-licensed software to create their own software product, then under the GPLv2, the company is required to disclose the source code of their software product when the product is distributed, if their software product constitutes a derivative work).

Unfortunately, the GPLv2 license does not define what constitutes a derivative work; instead, the GPLv2 reads: “work based on the Program means either the Program or any derivative work under copyright law”.⁵⁵ Therefore, the authors of the GPLv2 assumed that courts around the world would interpret the term “derivative work” under their local copyright law to determine what constitutes a derivative work. However, in the remainder of the license text, the term “derivative work” is mentioned in four more places with inconsistent references and implications to its meaning. The result of this is that the interpretation of the term “derivative work” is much broader in the license than under most national copyright laws around the world.

⁵⁰ Section 1 GPLv2.

⁵¹ Section 2(a) GPLv2.

⁵² Section 2(b) GPLv2.

⁵³ Section 3 GPLv2.

⁵⁴ Section 4 in conjunction with section 6 GPLv2.

⁵⁵ Section 0 GPLv2.

As there is no judicial opinion on the derivative work question, it is best to look at how the community interprets the license and what the intent of the authors was. The FSF has an extensive FAQ-page on its website where the following scenarios are considered a derivative work: (i) static or dynamic linking⁵⁶, (ii) plug-ins that are linked to a GPL-licensed program⁵⁷, (iii) different modules that are included in the same executable file⁵⁸, (iv) modules that are designed to run linked together in a shared address space.⁵⁹ Furthermore, organizations that are independent from the FSF interpret the meaning of “derivative work” similarly.⁶⁰

4.2.4.2 GPLv3

The GPLv3 license was released 15 years after the second version of the license. In spirit, the license is very similar to the GPLv2 license but there are a few noteworthy changes to list. First, the term “distribution” has been replaced with “to propagate” and “to convey”. These new terms are defined in the new definitions section of Section 1, and state:

To propagate a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or making modifications that you do not share.

*To “convey” a work means any kind of propagation that enables other parties to make or receive copies. **Mere interaction** with a user **through a computer network**, with no transfer of a copy, **is not conveying**.*

Therefore, following from these two terms, you are free to propagate the work as much as you want as long as no third party gains access to the work. If, however, a third party does have access to the work, then you are conveying the work and from that moment the conditions under the GPLv3 kick in. In practice, this means that a company is free to propagate a GPLv3

⁵⁶ See: <https://www.gnu.org/licenses/gpl-faq.html#GPLStaticVsDynamic> (retrieved February 18, 2017).

⁵⁷ See: <https://www.gnu.org/licenses/gpl-faq.html#GPLAndPlugins> (retrieved February 18, 2017).

⁵⁸ See: <https://www.gnu.org/licenses/gpl-faq.html#MereAggregation> (retrieved February 18, 2017).

⁵⁹ Ibid.

⁶⁰ These organizations include the Software Freedom Conservancy < <https://sfconservancy.org> > and the Software Freedom Law Center < <https://www.softwarefreedom.org> > (retrieved February 18, 2017).

license internally as much as it wants without being obliged to comply with the disclosure of the source code condition.

It is also interesting to note that the GPLv3 license explicitly states that “interaction through a computer network” does not constitute conveying. Therefore, also under the GPLv3, SaaS use does not make the conditions under the GPLv3 license active.

Furthermore, the term “derivative work” is no longer included in the GPLv3 license. Instead, the license speaks of modification and defines it as:

To modify a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.⁶¹

Unfortunately, this definition still does not make clear when someone is required to license additions made to the GPLv3 licensed program under the same GPLv3 license (i.e. what is a “derivative work”?). However, section 1 GPLv3 reads as follows:

*[...] For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for **shared libraries and dynamically linked subprograms** that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.*

Under section 5(c) GPLv3 you are required to:

[...] license the entire work, as a whole, under this License [...]

Therefore, GPLv3 explicitly states that (dynamically) linked programs to a GPLv3 licensed work must be licensed under the GPLv3 license as well.

⁶¹ Section 0 GPLv3.

What is also interesting about the GPLv3 is that it contains a patent grant clause⁶² that is very similar to the patent grant clause found in the Apache 2.0 License. This new clause is also one of the reasons why the Apache 2.0 License is compatible with the GPLv3, but not with the GPLv2 license.

4.2.4.3 AGPLv3

The AGPLv3 license is considered an ultra-strong copyleft license. This license is basically the same as the GPLv3 license but has one additional section:

*[...] your modified version must prominently offer all users interacting with it remotely through a computer network [...] an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge [...].*⁶³

Therefore, under the AGPLv3 license, you are required to disclose the source code of the software even if the software is distributed through a SaaS solution only. Note, however, that this condition only exists when you modify AGPLv3-licensed software; if you run an exact copy of the AGPLv3 licensed software, then you are not required to offer the source code to the users of the web-application.

4.2.4.4 LGPLv2.1 and 3

The L(esser)GPLv2.1 license is essentially the GPLv2 license with a few modifications and the LGPLv3 license is essentially the GPLv3 license with a few modifications. Both licenses are considered weak copyleft licenses with software libraries in mind.

Under the LGPL, a user is only obliged to release the source code of the library files. Therefore, if you build an application around a LGPL-licensed library, then you are not obliged to release the source code of your program that is linked to the library. However, you are still obliged to disclose your modification of the library, and it must be possible for any

⁶² Section 11 GPLv3.

⁶³ Section 13 AGPLv3.

user of your software to modify, recompile or replace the LGPL-licensed library and use its modified version with your software.⁶⁴

⁶⁴ Section 6 LGPLv2.1, section 4 LGPLv3.

4.3 Compatibility of Licenses

In essence, every open source license grants every user of the program an unrestricted right to use, modify, and distribute the software. However, not every open source license has the same conditions that users must comply with. Generally, permissive licenses have few conditions that a user must comply with. In fact, in most cases permissive licenses only have notice requirements. Copyleft licenses, on the other hand, have more stringent conditions.⁶⁵

It is best to explain what compatibility of licenses entails by providing examples. If, for example, you find open source software licensed under the GPLv2 license and you take this software and modify it by adding a few new software components, then under the copyleft provisions of the GPLv2 license, you are obliged to license the work as whole (i.e. the original licensed software with your modification added to it) under the GPLv2 license.

In another example, one of the components that you added to the GPLv2 licensed software was also an open source licensed component. However, this component was licensed under the MIT license and not GPLv2. What would happen in this circumstance? As noted in the previous paragraphs, the MIT license basically has just one condition (i.e. an attribution and notice requirement). The GPLv2 license, on the other hand, has far more conditions that need to be complied with to be able to use the software. However, one of the GPLv2 conditions is also the attribution and notice requirement. Another condition is the requirement to license a software program that includes GPLv2-licensed software under the same rights and conditions (i.e. under the same license).

Therefore, one can bundle MIT Licensed software with GPLv2 licensed software and license the combination under GPLv2 without any license incompatibility issues. After all, under the MIT License you are only required to give attribution to the author, and give notice that the software is provided on an “as-is” basis without warranty. Those are conditions that a GPLv2 license also has. Therefore, when you license the combination of the work under GPLv2, you automatically comply with both the MIT License conditions and the GPLv2 license conditions. It must be clear, however, that it is not possible to license the combination of the work under the MIT License because, by doing so, the conditions of the GPLv2 license would

⁶⁵ Meeker, H. (2015). *Open source for business: a practical guide to open source software licensing*. 1st ed. North Charleston, SC: CreateSpace Independent Publishing Platform, p. 59.

be violated. Therefore, although the MIT License is compatible with the GPLv2 license the GPLv2 license is not compatible with the MIT License.

4.3.1 Open Source Licenses/Proprietary Licenses

Generally, every software program that is licensed under permissive open source licenses may be used in a closed source environment and may be licensed under a proprietary license. On the other hand, it is generally not possible to use GPL-licensed software in a closed source environment, or license the software under a proprietary license. The GPL licenses contain the following condition “you may not impose any further restrictions on the exercise of the rights granted or affirmed under this License”.⁶⁶ If you create a software program that incorporates GPL-licensed software, then you are not allowed to impose any further restrictions on the recipient of the software that are not already covered in the GPL.

Proprietary licenses often contain restrictions such as the prohibition to make copies, decompile the object code, or the prohibition to make use of the software for certain purposes. These restrictions can be considered as further restrictions. Licensing a program containing GPL-licensed code under such a proprietary license would, therefore, constitute violation of the GPL license terms.

⁶⁶ Section 6 GPLv2 and Section 10 GPLv3.

5. Legal Approach to Open Source Licenses

5.1 Applicable Law and Jurisdiction

All open source licenses contain terms and conditions to which a user is deemed to have agreed. However, none of the licenses covered in this article have a so-called “governing law” or a “jurisdiction” clause. Therefore, one could wonder which law would apply to the license, and which courts would have jurisdiction if a legal dispute arose. Unfortunately, there has been no case law on this matter to shed light on these issues.

In the European Union, however, the Rome I Regulation⁶⁷ specifically deals with the question of which law governs a contract. The main principle of the Rome I Regulation is that parties are free to choose the law that should govern their contract.⁶⁸ However, in the absence of choice, article 4 Rome I Regulation applies; more specifically, paragraph 2 of article 4 applies. This paragraph states that the law that should govern the contract should be the law “of the country where the party required to effect the characteristic performance of the contract has his habitual residence.” In the case of an open source license, it can be argued that a license agreement is distinct from other contracts by the performance of the person granting the license.⁶⁹ After all, without the grant of the licensor, no one would be able to make use of the code without potentially infringing the author’s copyrights.⁷⁰ This means that, based on the Rome I Regulation – which is applicable law in all member states of the European Union – the law of the country where the licensor resides should be the law governing the open source license.

There is, however, one complication in the case of open source licenses. Open source projects are characterized by the fact that hundreds, if not thousands of contributors all over the world can contribute source code to a project. One can wonder in those cases, then, which law should apply if there is a plurality of authors residing all over the world. Unfortunately, there has been no case law on this rather specific, yet important, question on open source licenses.

⁶⁷ Regulation (EC) No 593/2008 of the European Parliament and of the Council of 17 June 2008 on the law applicable to contractual obligations (Rome I).

⁶⁸ Article 3(1) Rome I Regulation.

⁶⁹ Mireille M.M. van Eechoud, *Choice of Law in Copyright and Related Rights, Alternatives to the Lex Protectionis*, Kluwer Law International, The Hague/London/New York, 2003, p. 198ff.

⁷⁰ In many countries, certain uses of copyright protected work is permitted under certain conditions, e.g. using copyright protected work for educational purposes, or using copyright protected work under the fair-use principle found in common law countries.

The question of which courts have jurisdiction in case of legal disputes surrounding the open source license, one could look at the Brussel I bis-Regulation (also known as the EEX-regulation)⁷¹, which is applicable law in all the member states of the European Union. In the case of open source licenses, there are two possibilities based on the Brussel I bis-regulation.

(i) The main principle is that the court in the member state of the party that gets sued has jurisdiction.⁷² (ii) However, parties are free to contractually depart from this main principle.⁷³

In the case of open source licenses, this has not happened as there is no “jurisdiction” clause in any of the popular licenses. This means that legally, in the European Union, the court in the member state of the party that gets sued has jurisdiction.

⁷¹ Regulation (EU) No 1215/2012 of the European Parliament and of the Council of 12 December 2012 on jurisdiction and the recognition and enforcement of judgments in civil and commercial matters.

⁷² Article 4(1) Brussel I bis-Regulation.

⁷³ Article 25 Brussel I Bis-regulation.

5.2 Enforcement of Open Source License

Enforcement of open source licenses is not something that has happened often. This is probably due to the fact that open source licenses are, in essence, copyright licenses. This means that only copyright holders in an open source project have standing to start a case against an alleged open source license violator.⁷⁴ In addition, many open source projects are created by individual developers. Historically, legal enforcement of open source licenses has been low amongst these developers because they would have to engage attorneys and/or other professionals to assist them in ensuring license compliance. Furthermore, what also makes enforcement of open source licenses more difficult is the fact that open source projects usually have more than one author. In some cases, there could be hundreds, if not thousands, of contributors to an open source project. In these instances, it can be difficult for developers to prove that they hold copyright on the work in the first place.⁷⁵

If, however, someone is ever confronted with a lawsuit concerning an open source license, there are a few likely scenarios. The first thing a court will probably do, is assess whether the plaintiff is a or the copyright holder in the open source project in dispute. If not, then it is likely that the case will be dismissed as the plaintiff will have no standing to sue someone over alleged copyright infringement.

If, however, the plaintiff is found to be a copyright holder in the open source project, then the court will determine whether the defendant has violated the license terms. If the defendant has indeed violated the license terms, then the court will determine whether the defendant has infringed the plaintiff's copyright. It can be possible that the defendant was indeed violating the license but was not infringing the plaintiff's copyright (for example, by not using any of plaintiff's code). If that is the case, then the plaintiff would probably have no standing to go further with the lawsuit as his copyright would not have been infringed.

If, however, the court finds that the defendant was both violating the license and subsequently infringing the plaintiff's copyright, then the defendant would be ordered to do one or more of the following: (i) stop infringing the plaintiff's copyright (e.g. stop making use of the code),

⁷⁴ Meeker, H. (2015). *Open source for business: a practical guide to open source software licensing*. 1st ed. North Charleston, SC: CreateSpace Independent Publishing Platform, p. 59.

⁷⁵ See Hellwig / VMware Global Inc. for example.

(ii) compensate the plaintiff (legal fees and/or damages, if any, amount depends on the applicable law and jurisdiction), and/or (iii) act in compliance with the open source license (e.g. making the source code of the program available under the GPL license).

6. Case Law Open Source Licenses

6.1 The Netherlands

In the Netherlands, there is no case law concerning open source licenses. However, in 2006 the Curry/Audax case⁷⁶ was the first case ever in the world concerning the Creative Commons (CC) licenses. The CC licenses are licenses used for artistic works such as pictures, writings or videos. There are several types of CC licenses; however, every CC license grants the baseline right to distribute the copyrighted work worldwide for non-commercial purposes.

6.1.1 Curry/Audax (Amsterdam, 2006)

Adam Curry (Dutch celebrity) hosted pictures of himself and his family on the picture-sharing website Flickr. These pictures were licensed under the CC license.⁷⁷ In 2006, a Dutch tabloid republished four of these pictures in a story about Curry without Curry's permission. Curry sued the tabloid and argued that the tabloid had violated the CC license by using the photos in a commercial magazine, and by not including a reference to the CC license itself. The tabloid, on the other hand, argued that it thought that the photos could be taken and used because they were marked as "public" on Flickr. The court held that the conditions of a CC license automatically apply to the content licensed under it, and binds users of such content even without expressly agreeing to, or having knowledge of, the conditions of the license.⁷⁸ This meant that the tabloid had indeed violated the CC license.

Although not specifically about open source licenses, this case is still important because it demonstrates that CC licenses are enforceable in court under copyright law. This means that a court would probably rule the same way in a case concerning open source licenses.

⁷⁶ Rb. Amsterdam (vzr.) 9 maart 2006, ECLI:NL:RBAMS:2006:AV4204.

⁷⁷ More specifically, they were licensed under the CC BY-NC-SA 2.0 license, meaning that users of the pictures were obliged to give attribution, not use the pictures for commercial purposes, and only redistribute the pictures under the same conditions.

⁷⁸ Paragraph 5 of the ruling.

6.2 Elsewhere in the European Union

There is little case law concerning open source licenses in the rest of the European Union. There are, however, several cases in other member states of the European Union concerning the CC licenses where the result was similar to that in the Curry/Audax case (i.e. the CC license was found to be valid and enforceable, and in most of the cases the court found that the defendant had violated the license).⁷⁹

There is, however, some case law concerning open source licenses in Germany, and that is mainly due to one person: Harald Welte. Welte is one of the main programmers of the Linux netfilter/iptables project and in 2004 started to sue companies in Germany that violated the GPLv2 license under which netfilter/iptables was licensed.⁸⁰

6.2.1 Welte/Sitecom (Munich, 2004)

Sitecom, a company selling routers, used the netfilter/iptables software in its routers. However, Sitecom did not provide the source code of the program, nor did it refer to the GPLv2 license (both conditions under the GPLv2 license). Welte, author of the netfilter/iptables software, filed an injunction against Sitecom. Sitecom argued that it had not accepted the GPLv2 license and thus was not bound by its terms. The court ruled that Sitecom was, in fact, bound by the GPLv2 license, as Sitecom would have no right in the first place to make use of the netfilter/iptables software without a proper license. Sitecom was ordered to comply with the GPLv2 license terms or stop selling the routers making use of the netfilter/iptables software. Many of the cases that followed the Sitecom case were very similar in both action taken and the result achieved.⁸¹

6.2.2 Welte as a third party in AVM/Cybits (Berlin, 2011)

AVM is the manufacturer of the FRITZ!Box routers, which are very popular routers in Germany. These routers make use of the netfilter/iptables software authored by Welte. Cybits, on the other hand, is a software company that offered internet filtering software. To use the

⁷⁹ See for example Le Tribunal de première instance de Nivelles, 2010-10-26, Case No. 09-1684-A. (Allard et al / Festival de Theatre de Spa), Landgericht Berlin, 8-10-2010, Case No. 16 O 458/10 (Gerlach / DVU), and Svea hovrätt, 25-02-2016, (Äventyrsgruvan i Tuna Hästberg v. Gunnarsson).

⁸⁰ See <gpl-violations.org> (retrieved February 18, 2017).

⁸¹ These cases were: Welte / Fortinet UK Ltd. (Munich, 2005), Welte / D-Link (Frankfurt, 2006), and Welte / Skype (Munich, 2008).

filter, Cybits sold a program that downloaded the FRITZ!Box firmware, modified it, and installed the modified firmware directly on the router. AVM was not pleased with this and applied for a preliminary injunction against Cybits. AVM justified its request for a preliminary injunction based on an alleged infringement of copyright. The injunction was granted by the court, which Cybits appealed.

It was then, when Welte became aware of the case and joined the legal dispute as an intervener. Welte claimed that AVM's claim was illegal because they were bound by the GPLv2 license conditions, as the firmware was making use of the netfilter/iptables software. By prohibiting Cybits from modifying the source code of the firmware, AVM was violating the terms of the GPLv2 license, viz., allowing anyone to use, make copies of, and modify etc. the source code.

Therefore, the court held that Cybits was not infringing AVM's copyright, as AVM was bound by the GPLv2 license terms.

6.2.3 Welte/Fantec (Hamburg, 2013)

Fantec, a manufacturer of consumer electronics, was selling a media player that incorporated the netfilter/iptables software, authored by Welte, in the firmware of one of its media players. On Fantec's website, people could download the source code of the firmware, and thus, it looked like Fantec was complying with its conditions under the GPLv2 license. However, Welte discovered that the source code of the firmware provided by Fantec did not correspond with the firmware used by the media player. Under the GPLv2 license, you are required to make available the exact source code used by a product.

Welte sued Fantec for violation of the GPLv2 license and argued that Fantec was providing an outdated and incomplete version of the firmware's source code. Fantec, on the other hand, argued that it could not be held responsible for the violation because the source code of the firmware was provided by one of their Chinese contractors with assurance of software integrity.

The court held that Fantec was violating the GPLv2 license terms, and held that it is the seller's own responsibility to determine whether the source code provided by a contractor is indeed the source code used in the products sold by the seller.

6.2.4 Hellwig/VMware Global Inc. (Hamburg, 2016)

In 2015, Christoph Hellwig sued VMware Global Inc. for violation of the GPLv2 license. Hellwig is one of the many developers of the Linux kernel and has contributed thousands of lines of code to the Linux kernel project.⁸² VMware, on the other hand, is a software company that, among many other programs, offers a program called VMware ESXi. VMware is using parts of the Linux kernel, which are licensed under GPLv2, in their closed source ESXi product. The Linux parts used by VMware have been completely modified and are integrated in a kernel called “vmkernel”. The source code of the vmkernel kernel has been released under the GPLv2 license by VMware.

However, the ESXi product is, at its core, driven by a kernel called “vmkernel”. This kernel dynamically links to the vmkernel kernel. VMware has not released its source code for the vmkernel kernel. VMware argues that vmkernel should not be considered as a derivative work of the Linux kernel and therefore the copyleft terms of the GPLv2 license do not apply to the vmkernel. Hellwig, on the other hand, argues that vmkernel is a derivative work of the Linux kernel and therefore, VMware should either abide by the GPLv2 license terms (making the source code available) or stop using the Linux kernel.

When this case was brought forward by Hellwig, it looked like we would finally get a judgement ruling on the question of what constitutes a derivative work under the GPLv2 license. However, the case was dismissed in July 2016 by the Hamburg District Court.

Under (German) copyright law, only authors of a copyrighted work can sue for (alleged) copyright infringement. The Linux kernel is an open source project that consists of millions of lines of code. Those lines have been written by thousands of developers all over the world. In this particular case, Hellwig had not convinced the judge that the code used by VMware was indeed code written by Hellwig himself. Therefore, VMware was not found guilty of infringing Hellwig’s copyright, nor was it found guilty of violating the GPLv2 license by not releasing the source code of the vmkernel kernel, because the court did not get to that

⁸² Software Freedom Conservancy. (n.d.). Frequently Asked Questions about Christoph Hellwig's VMware Lawsuit. Retrieved February 18, 2017, from <<https://sfconservancy.org/copyleft-compliance/vmware-lawsuit-faq.html>>.

question in the case. In August 2016, Hellwig appealed the decision. It is therefore possible that we will see a new ruling on this matter later this year (2017).

6.3 United States

There has been some litigation over open source licenses in the United States. However, most lawsuits concerning open source license issues are settled out of court. This means that it is still unclear how open source licenses should be interpreted in the US. The two cases discussed below are the most important cases concerning open source licenses in the US.

6.3.1 Jacobsen v. Katzer (9th Cir., 2006–2010)⁸³

The Jacobsen v. Katzer case was the first big win regarding open source license enforcement for the open source community in the US. In short, this case was about the question of whether an open source license should be considered as a contract or as a license. This distinction is, apparently, very important for the legal enforceability of open source licenses in the US.

At issue in this case was the rarely used Artistic License. Jacobsen had written source code and licensed the code under the Artistic License. Katzer had taken Jacobsen's code and removed all the original copyright notices to the original authors and substituted them with Katzer's company name. Jacobsen argued that because Katzer had used the code in violation of the license, his use of the code constituted copyright infringement. However, the District Court held that the Artistic License was a contract between the parties and remedies for breach of the contract could only occur in contract law (i.e. damages as determined under state law) and not as a copyright infringement action (statutory damages, and injunctive relief under federal law).

The ruling of the District Court was a big blow to the open source community. Therefore, Jacobsen was supported by many third parties⁸⁴ during his appeal at the Court of Appeals for the Federal Circuit (CAFC). Fortunately, for both Jacobsen and the open source community, the CAFC found that the Artistic License was a conditional license and not a contract. This meant that Jacobsen could seek remedies for violation of copyright rather than the more limited remedies available under contract law. The CAFC remanded the case to the District Court to determine if the other criteria for injunctive relief had been met. However, before the

⁸³ To see the pleadings and rulings in this case, see: <http://jmri.sourceforge.net/k/docket/index.shtml> (accessed 12 February 2017).

⁸⁴ Organizations such as The Linux Foundation, The Open Source Initiative, and the Software Freedom Law Center filed an amicus brief before the CAFC in support of Jacobsen.

District Court could give its judgement on this matter, Jacobsen and Katzer had reached an out-of-court settlement.

This case is seen as the most important case, to date, concerning open source licenses in the US is for two reasons. First, the CAFC confirmed that open source licenses are enforceable in a court of law and second, the CAFC confirmed that open source licenses should be seen as a conditional copyright license rather than a contract. This is important because plaintiffs have more remedies under federal copyright law than under state contract law.

6.3.2 Busybox cases (S.D.N.Y., 2007–2012)

In 2007, the Software Freedom Law Center (SFLC) filed a copyright infringement lawsuit against Monsoon Multimedia, Inc. This lawsuit was filed on behalf of Erik Andersen and Rob Landley, two main developers of Busybox, a GPLv2 licensed software product, and is considered the first-ever US copyright infringement lawsuit based on a violation of the GPL license.⁸⁵

SFLC argued that Monsoon was using the Busybox software in their products without complying with the GPLv2 license terms. The GPLv2 license requires that each recipient of GPL-licensed code is provided access to the source code of the GPL-licensed program. In this case, Monsoon was selling products with the Busybox software embedded in them. However, Monsoon did not offer their customers access to the source code of the Busybox software.

In less than two months after the lawsuit was filed, the parties reached an out-of-court settlement. Monsoon agreed to release the source code of their modified version of Busybox. Furthermore, they agreed to appoint an Open Source Compliance Officer to monitor and ensure GPL compliance and they agreed to pay an undisclosed amount of “financial consideration” to the plaintiffs.⁸⁶

⁸⁵ Software Freedom Law Center. (2007, September 20). On Behalf of BusyBox Developers, SFLC Files First Ever U.S. GPL Violation Lawsuit. Retrieved February 18, 2017, from <<http://www.softwarefreedom.org/news/2007/sep/20/busybox/>>.

⁸⁶ Software Freedom Law Center. (2007, October 30). BusyBox Developers and Monsoon Multimedia Agree to Dismiss GPL Lawsuit. Retrieved February 18, 2017, from <<http://www.softwarefreedom.org/news/2007/sep/20/busybox/>>.

After this first copyright infringement case concerning the Busybox software, many other lawsuits were filed by the SFLC on behalf of the two developers.⁸⁷ Almost all cases ended the same way (i.e. the cases were settled and the defendants either agreed to stop using the software or release their modifications of the software). In most cases the defendants agreed to appoint an Open Source Compliance Officer and they agreed to pay an undisclosed sum of money to the plaintiffs.⁸⁸

What is interesting about these cases is that Rob Landley, one of the developers represented by the SFLC later stated that he regretted that he took part in the Busybox cases. Landley hoped that, by enforcing the GPL license, the companies sued would comply with their license conditions and be more engaged with the open source community. Instead, what happened was the exact opposite. The companies sued would indeed comply with the license terms and provide their version of the source code. However, none of the code provided by the defendants turned out to be useful at all. Furthermore, many companies decided to completely stop using any software licensed under the GPL. This meant that instead of having more companies engaged with the open source community, more and more companies stopped using GPL-licensed software altogether because of the compliance risks attached to it.⁸⁹

⁸⁷ These included lawsuits against, among others, Verizon, Best Buy, Western Digital, Samsung, and JVC.

⁸⁸ See: <<https://www.softwarefreedom.org>> for more information.

⁸⁹ Proffitt, B. (2012, February 02). GPL enforcement sparks community flames. Retrieved February 18, 2017, from <<http://www.itworld.com/article/2732025/open-source-tools/gpl-enforcement-sparks-community-flames.html>>.

7. How to Comply with the Open Source Licenses?

Legend

Private use: software may be used privately. This includes using the software commercially within one organization. By using the software privately, the conditions do not apply.

Commercial use: software may be used for commercial purposes e.g. selling the software.

Modify: software may be modified.

Distribute: software may be distributed. When the software is distributed, the conditions apply. Note: SaaS / ASP use is not deemed as distribution.

Copyright notice: notice of the copyright(s) must be included when distributing the software.

Include license: a copy of the applicable license must be included when distributing the software.

Warranty: the license explicitly states that it does not provide any warranty. Note: you are free to deviate from this limitation on a contractual basis.

Liability: the license includes a limitation of liability.

Jurisdiction: jurisdiction of the court(s).

Applicable law: which law should govern the license.

Trademark use: trademarks included in the software may, or may not be used by the licensee.

Patent use: patents included in the software may, or may not be used by the licensee.

Source disclosure: The source of the software must be disclosed when the software is distributed.

Same license: Modifications must be released under the same license when distributing the software (copyleft).

7.1 MIT License

Rights ¹	Conditions ²	Limitations ³	Topics not covered
• Private use • Commercial use • Modify • Distribute	• Copyright notice • Include license	• Warranty • Liability	• Jurisdiction • Applicable law • Trademark use • Patent use • Source disclosure • Same license

1. These rights are explicitly granted under paragraph 1 of the license.
2. These conditions are explicitly set out under paragraph 2 of the license.
3. These limitations are explicitly covered under paragraph 3 of the license.

7.2 BSD 2- & 3-clause License

Rights ¹	Conditions ²	Limitations ³	Topics not covered
• Private use • Commercial use • Modify • Distribute	• Copyright notice • Include license	• Warranty • Liability • Trademark use⁴	• Jurisdiction • Applicable law • Patent use • Source disclosure • Same license

1. These rights are explicitly granted under paragraph 1 of the license.
2. These conditions are explicitly set out under the 1st and 2nd clause of the License.
3. These limitations are explicitly covered under the last paragraph of the license.
4. Trademark use is explicitly prohibited under the 3-clause version of the BSD License only. This limitation is found under clause 3 of the BSD 3-clause License. However, even though the 2-clause version does not contain this explicit prohibition on the use of the ‘‘name of the copyright holder nor the names of its contributors’’, it is safe to assume that the 2-clause version does not grant you an implicit right to use those names, either.

7.3 Apache 2.0 License

Rights ¹	Conditions	Limitations	Topics not covered
• Private use • Commercial use • Modify • Distribute • Patent use²	• Copyright notice⁵ • Include license³ • State changes⁴ • Include notice⁶	• Warranty⁷ • Liability⁸ • Trademark use⁹	• Jurisdiction • Applicable law • Source disclosure • Same license

1. Most of these rights are explicitly granted under section 2 of the License.
2. The right to use patents is covered under section 3 of the License.
3. This condition is explicitly set out under section 4(a) of the License.
4. This condition is explicitly set out under section 4(b) of the License.
5. This condition is explicitly set out under section 4(c) of the License.
6. This condition is explicitly set out under section 4(d) of the License.
7. This limitation is explicitly set out under section 7 of the License.
8. This limitation is explicitly set out under section 8 of the License.
9. This limitation is explicitly set out under section 6 of the License.

7.4 GPL

7.4.1 GPLv2

Rights	Conditions	Limitations ⁹	Topics not covered
• Private use¹ • Commercial use¹ • Modify² • Distribute³	• Copyright notice⁴ • Include license⁵ • State changes⁶ • Source disclosure⁷ • Same License⁸	• Warranty • Liability	• Jurisdiction • Applicable law • Trademark use • Patent use

1. This right is granted under section 1 GPLv2.
2. This right is granted under section 2 GPLv2.
3. This right is granted under sections 1, 2 and 3 GPLv2.
4. This condition is set out under sections 1 and 2(c) GPLv2.
5. This condition is set out under section 1 GPLv2.
6. This condition is set out under section 2(a) GPLv2.
7. This condition is set out under section 3 GPLv2.
8. This condition is set out under section 2(b) and 4 GPLv2.
9. These limitations are found under sections 11 and 12 GPLv2.

7.4.2 GPLv3

Rights	Conditions	Limitations ⁹	Topics not covered
• Private use¹ • Commercial use¹ • Modify² • Distribute³ • Patent use⁴	• Copyright notice⁵ • Include license⁵ • State changes⁶ • Source disclosure⁷ • Same License⁸	• Warranty • Liability	• Jurisdiction • Applicable law • Trademark use

1. This right is granted under section 2 GPLv3.
2. This right is granted under section 4 GPLv3.
3. This right is granted under sections 4, 5 and 6 GPLv3.
4. This right is granted under section 11 GPLv3.
5. These conditions are set out under sections 4 and 5 GPLv3.
6. This condition is set out under section 5(a) GPLv3.
7. This condition is set out under section 6 GPLv3.
8. This condition is set out under sections 4, 5 and 6 GPLv3.
9. These limitations are found under section 15, 16 and 17 GPLv3.

Appendix A – Rights, Conditions and Limitations

License	Private Use	Commercial Use	Modify	Distribute	Copyright Notice	Include License	Warranty	Liability	Trademark Use	Patent Use	Source Disclosure	Same License
MIT	✓	✓	✓	✓	✓	✓	X	X				
BSD 2&3-clause	✓	✓	✓	✓	✓	✓	X	X	X ²			
Apache 2.0	✓	✓	✓	✓	✓	✓	X	X	X	✓		
GPLv2	✓	✓	✓	✓	✓	✓	X	X			✓	✓
GPLv3	✓	✓	✓	✓	✓	✓	X	X		✓	✓	✓
LGPLv2.1/3	✓	✓	✓	✓	✓	✓	X	X			✓ ³	✓ ⁴
AGPLv3	✓	✓	✓	✓ ¹	✓	✓	X	X		✓	✓	✓

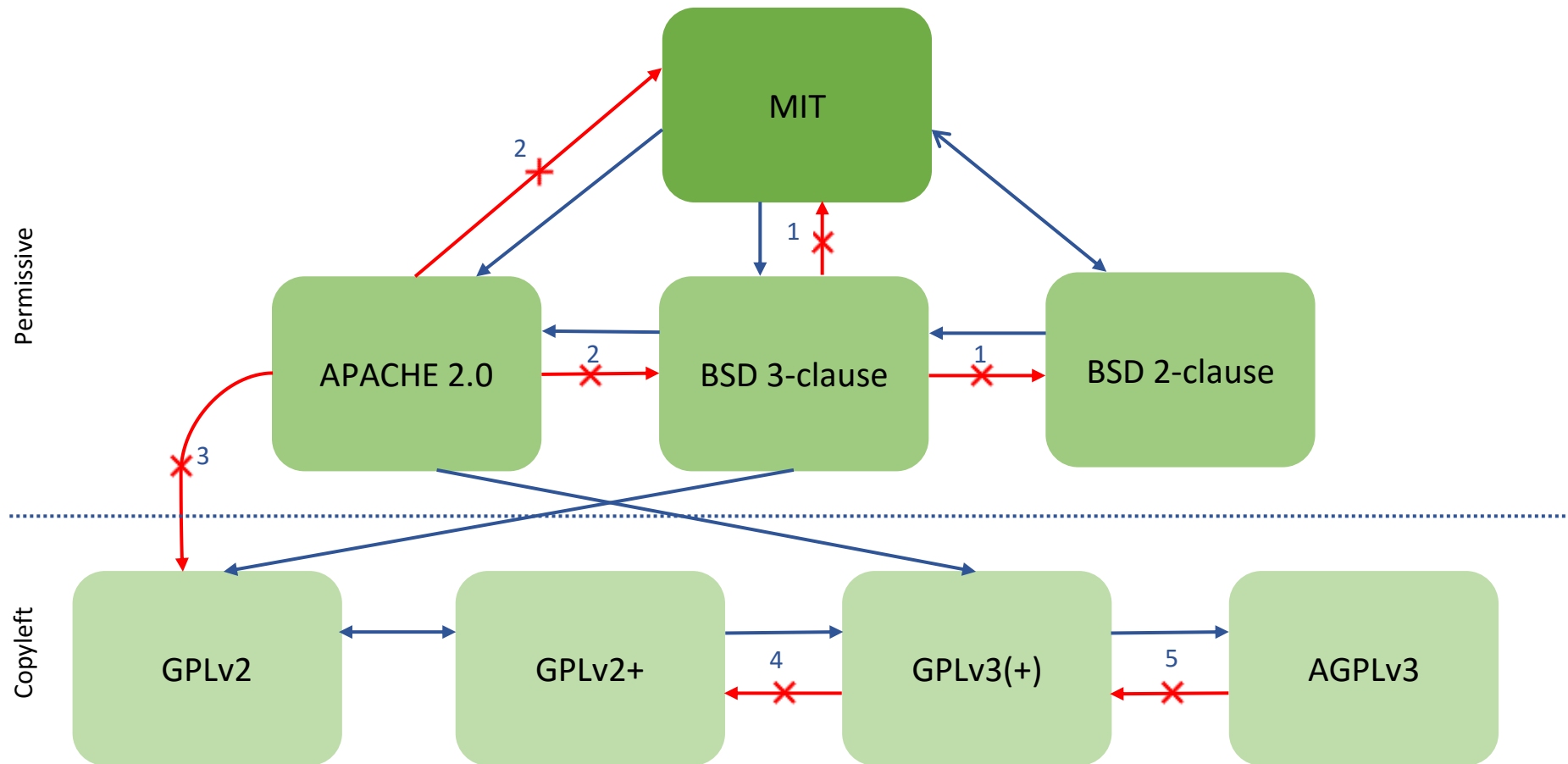
✓ = Right

✓ = Condition

X = Limitation

1. Network distribution (ASP/SaaS) constitutes distribution (section 13 AGPLv3)
2. Only the BSD 3-clause License has a clause that could be interpreted as a prohibition to use any trademark rights (3rd clause BSD 3-clause License). However, it is generally understood that the BSD 2-clause License (or any other open source license for that matter) does not grant an implicit trademark right either.
3. If your own software is dynamically linked to the LGPL licensed software, then only the source code of the LGPL licensed software has to be disclosed.
4. If your own software is dynamically linked to the LGPL licensed software, then your own software does not have to be licensed under LGPL.

Appendix B – Compatibility of Open Source Licenses



1. The BSD 3-clause License has one additional condition and is therefore not compatible with the BSD 2-clause license and/or the MIT License.
2. The Apache 2.0 License has more conditions than the MIT and/or BSD 3-clause License.
3. See: <<https://www.apache.org/licenses/GPL-compatibility.html>> and <<https://www.gnu.org/licenses/license-list.en.html#apache2>>.
4. See: <<https://www.gnu.org/licenses/license-list.en.html#GPLv2>> and <<https://www.gnu.org/licenses/gpl-faq.html#AllCompatibility>>.
5. See: <<https://www.gnu.org/licenses/license-list.en.html#AGPLv3.0>>

