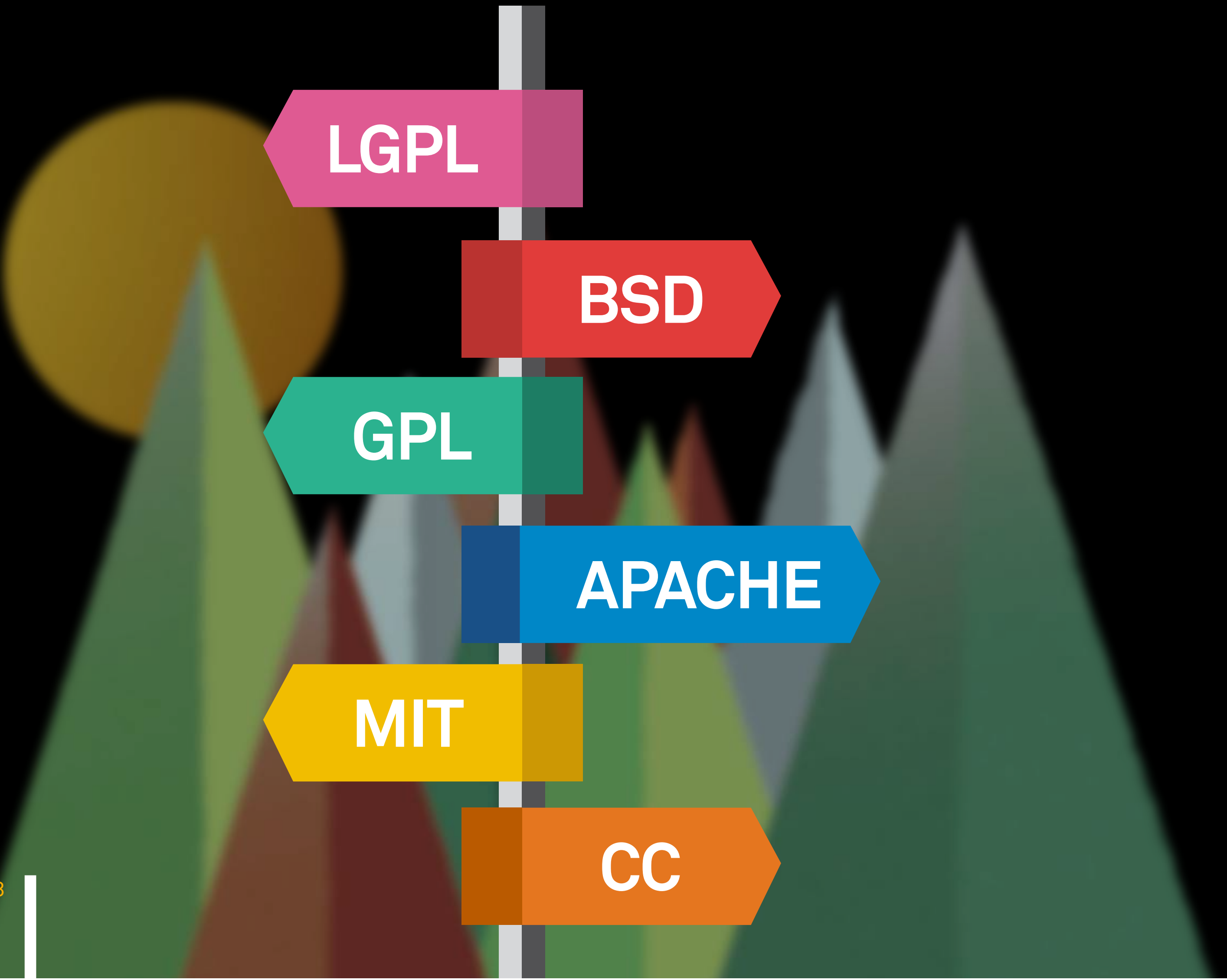
NAVIGATING THE LANDSCAPE OF LICENCES

Shashank Sharma has spent countless hours studying the various free and open source licences available to help you choose the best one for your project



The success of open source software and the impact it has had on everyday life is indisputable. From powering desktops of home users and providing cost-effective technology solutions to small and medium-sized businesses, open source has spread into powering large servers, running entire corporations and even analysing big data. But what is open source? How is it different from free software? Why are there so many software licences and how do you decide which is best for what situation?

These questions have plagued users and developers for decades and despite the large amount of detailed documentation, several scholarly articles and a book or two, the doubts persist. Part of the reason for this is the FUD campaign (fear, uncertainty, doubt) of the 1980s and 1990s perpetrated by large proprietary software companies, most notably Microsoft, with the aim or retaining their dominance of the market. The strategic spread of misinformation, targeting open source software as well as the underlying software licence, was designed to hamper its adoption in favour of proprietary alternatives.

Open source software, thanks to the work of tireless campaigners and developers who consistently ship reliable and stable product, has long since left fears of its reliability and performance far behind. Unfortunately, it's picked up a rather more difficult and cumbersome problem: licence proliferation. But even that is a bit of a FUD, this time being spread by users and developers who feel overwhelmed by the choices on offer.

Free and open source licences can be used for more than just software. What began as a contrarian movement for software has spread its reach into various other catalogues of work such as artwork, images, music, prose and more, which necessitated the proliferation. What one must understand, however, is that a single open source licence isn't ideal, nor recommended, for every possible use case. Certainly, what works for software may not for a comic book or a music album.

This article will hopefully help clear some of the misconceptions. While our focus is on open source licences for software, this should help you decide the best licence for other projects as well. We will elaborate on the distinction between open source and free software licences further in the feature. But, unless specified otherwise, our use of the term 'open source licence' refers to the entire gamut of free and open source licences.

Understanding copyright

To describe it simply, without venturing into the myriad possibilities of work for hire, performance, adaptation and so on, the copyright laws describe certain exclusive rights reserved solely for the creators of a work. The work may be anything from text to sound recordings, video films, images, artwork and software.

Copyright owners can prevent others from copying, modifying or distributing their works. The end-user licence agreements (EULAs) on proprietary software, such as games and office suites, describe the various rights granted to those who purchase a copy of the software. A creator by definition holds the copyright over works created by him or her.

But this model of restricting rights to a single creator doesn't account for the Linux community, where projects such as the Linux kernel feature contributions from nearly 14,000 individual developers, all around the world, working at more than a thousand different companies. ▸

“Free and open source licences can be used for more than just software”

Choosing a licence

There are several important questions that you must answer before deciding which software licence to choose for your project. The first is whether the finished software or project will be used in-house, used only by a select few or made available to the public at large.

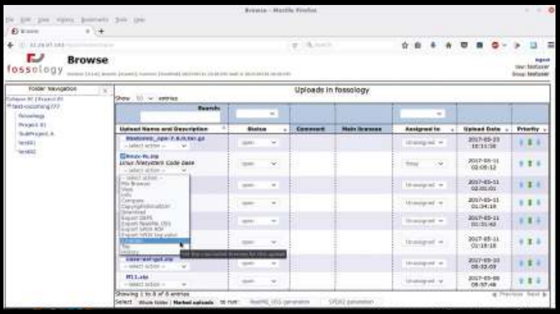
Before you decide to distribute the work to the public, also consider the effort required to market and show the utility of your software and to address bug reports and feature requests.

Having decided to publicly release the work, the next question is whether to release it as proprietary or not. Here, proprietary doesn't mean that the software is sold commercially. Similarly, non-proprietary doesn't necessarily imply that the work is distributed free of cost. You can also adopt the method used by Artifex Software which owns Ghostscript, and provides the software under a commercial licence as well as AGPL (formerly distributed under GPLv3). In fact, Artifex allows non-commercial licensees to freely use Ghostscript, provided all derivative software is also released under the same copyleft terms. It is this requirement that is at the root of the Artifex vs Hancom case, which we discuss later (see p31).

Should you decide on a non-proprietary model, you still have to make the decision to release the project under an open source licence or simply unleash it into the public domain. The latter signifies the waiver of all rights vested exclusively onto the creator. That is, once released into the public domain, the creator cannot initiate legal action claiming unsanctioned use of the software.

If after navigating the forks at each step, you decide to release your project under an open source licence, you're still confronted with yet another decision: whether to release the project under a copyleft licence or under an academic or open source licence.

The final decision will depend on your exact circumstances and further questions such as whether there are patents or other legal considerations.



Right FOSSology is a free scanner that will enable you to discover the licence used by a piece of software and help generate software copyright notices

Free software strives to protect and extend four freedoms to all users. These are: the freedom to use the software; study and modify the software; distribute the software and distribute the modifications.

A free software licence goes a step beyond the four freedoms as it includes an obligation which requires that all modifications, including derivative works, be distributed under the terms of the same free software licence. This is how the term 'copyleft' came about. It extends the rights reserved exclusively for creators and protected by copyright laws, to everyone.

Collaborative works

The insistence that derivative and modified works also be released under the same licence didn't gel with many in the community and this has led to a parallel movement and the term 'open source'.

The major point of difference between free software and open source is that the latter doesn't require release of derivative works under the parent licence, which is a strict requirement of copyleft licences. This is why open source licences, even when providing and ensuring the same freedoms as a free software licences, are not called copyleft licences.

The developer can use the terms of a licence to grant permissions to others. These permissions may be limited to using the software in very specific terms and may even define the circumstances, which is the case with most proprietary software licence agreements.

An open source licence can be further categorised as either reciprocal or academic. The reciprocal licences, also known as copyleft licences, are the ones that ensure that the freedoms available to a person or developer are also extended to all downstream users. That is, if a developer modifies a software released under a copyleft licence, they are required to also release the modifications under the same licence. The object of a copyleft licence is to ensure that innovations remain available to all downstream users and developers without any restrictions. The most popular copyleft licence is the GNU General Public License, commonly referred to as the GPL.

Academic licences – also called permissive licences – while also ensuring the freedom available to users, do away with restrictions entirely. This is similar to the imparting of knowledge at most universities. How that information is used and utilised is left entirely to the students. In a similar vein, academic licences such as the BSD, Apache and MIT provide carte blanche on the use case for software released under these licences, requiring only that the original creator be attributed.

Anatomy of a licence

The basis of copyleft and academic licences is in copyright law. The author of software is the licensor and all those who use the software are its licensees and must abide by the terms of the licence. The breach of the conditions imposed by a licence results in infringement of the copyright of the author of the software. By extending the exclusive rights to others by way of a licence, the original creator does not forfeit their copyright claims. This specific issue has been contended numerous times before the courts in various jurisdictions.

Licences and contracts are two distinct legal terms and must not be confused. A contract has several elements such as promise, offer, acceptance, consideration and so on. We don't have space to cover all these in detail, but we shall discuss two essential requirements of a contract:

- **Consensus ad idem:** Also known as meeting of the minds or mutual consent. It describes the mutual agreement of the contracting parties to be bound by the terms of the contract.

TOP TIP

A copyright infringement can result in the award of more damages than a breach of contract, which is best for enforcing specific terms.

GPL

The latest version, GPLv3 was published in 2007. Like its predecessor GPL, it requires the release of complete source code of licensed work. Modifications and derivative works should also be released under the same licence. Recommended only for software. Bash and GIMP are released under GPLv3.

MIT

A short and very permissive licence. Allows for licensed work to be used for commercial use. Derivative works may be released under different terms and without source code. Modifications can be used privately and don't have to be released publicly. Used by Rails, jQuery and many others for its simplicity.

BSD

The latest three-clause licence, while similar to the older BSD 2-clause and MIT licence, it makes an addition. It requires that derivative works not use the name of the original project or its developers for promotion without express permission. Only recommended for software and not other works.

• **Consideration:** It is the reason parties enter into a contract. It usually refers to a bargain arrived at by parties, wherein one party performs or agrees not to perform an act, in exchange of receiving a thing of certain value (for instance, money or goods).

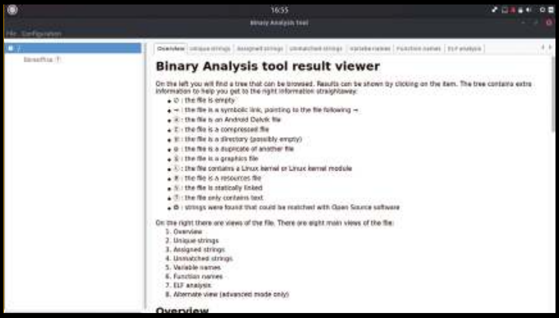
It has been argued that the mere use of the open source software implies an agreement to be bound by its terms and the courts have not taken a contrary position on this subject.

Unlike EULAs on proprietary software, open source licences do not require explicit consent from users before allowing them to use the software.

Since there is no apparent consent or exchange of consideration, there has long been doubt whether open source licences are in fact enforceable contracts. This is an important issue to understand because while there's no doubt that a breach of a condition imposed by an open source licence can result in a copyright infringement, contract law describes how contracts should be formed and interpreted, how to enforce them and the remedies for breach. Where a contract is silent on an important issue such as breach, the established contract law can also provide default terms.

In this respect, a recent ruling of the District Court of the Northern District of California, in the matter of Artifax Software Inc. vs Hancom Inc. refusing to dismiss a lawsuit alleging violation of an open source licence (GPL), is very promising.

This is because the defendant in the suit (Hancom) had pleaded that mere use of a software licensed under the GPL does not create a contract between the parties.



Binary Analysis tool result viewer

On the left you will find a tree that can be browsed. Results can be shown by clicking on the item. The tree contains extra information to help you get to the right information straightaway:

- ∅ : the file is empty
- → : the file is a symbolic link, pointing to the file following →
- Ⓐ : the file is an Android Dalvik file
- Ⓒ : the file is a compressed file
- Ⓓ : the file is a directory (possibly empty)
- Ⓓ : the file is a duplicate of another file
- Ⓖ : the file is a graphics file
- Ⓛ : the file contains a Linux kernel or Linux kernel module
- Ⓡ : the file is a resources file
- Ⓢ : the file is statically linked
- Ⓣ : the file only contains text
- ⊕ : strings were found that could be matched with Open Source software

On the right there are views of the file. There are eight main views of the file:

1. Overview
2. Unique strings
3. Assigned strings
4. Unmatched strings
5. Variable names
6. Function names
7. ELF analysis
8. Alternate view (advanced mode only)

Overview

However, the court held that GPL's terms explicitly require compliance of conditions for distribution of software. While the order is significant on many fronts, the ruling essentially means that apart from copyright infringement, the lawsuit can also proceed for breach of contract.

The Federal Circuit court had previously (in Jacobzen vs Katzer) held that the lack of money changing hands in open source software does not imply there is no economic consideration, which helped pave the way towards establishing open source licences as contracts.

Ensuring enforcement

For developers, it's important to understand that open source licences are enforceable. But how does one determine if a particular project violates an open source licence? Similarly for companies, it is important to make certain they are not inadvertently in breach of the terms of a licence.

Compliance engineers, as the name suggests, work towards identifying possible breaches of open source licences. They do so by meticulously analysing software using various tools. The popular and aptly named Binary Analysis Tool is used for studying binaries. Similarly, binwalk is popular for analysing and reverse-engineering firmware images.

Another option is FOSSology, a free licence scanner that serves several functions. It can be used to determine the software licences of a specified piece of software. It also provides a web-based interface and lets users generate copyright notices for their software and much more.

But if you're looking for more specific guidance in ensuring compliance, you can turn to organisations that are devoted to that specific purpose. Herald Welte's **gpl-violations.org** project has done tremendous work in the past to ensure compliance of GPL.

Closer to home, FSF Europe has long been active in the legal sphere; it hosts events, answers questions and provides other types of assistance on licensing issues, as well as running mailing lists to help users and developers. ■

TOP TIP

The use of '+' in licences such as GPL means that the software can be released under the later version of the licence if the developer so desires.

Left The Binary Analysis Tool ships a number of scripts and a GUI to assist in compliance efforts. We suggest you read through the documentation or you'll be entirely lost

CREATIVE COMMONS

The CC licences are recommended for non-software works such as images, artwork and music. The copyleft CC-BY-SA allows for modifications of licensed work, distribution of derivative work for commercial use, but under the same licence. CC-BY is permissive and only requires attribution.

APACHE LICENSE 2.0

Recommended for software if you want a permissive licence but also want to grant patent rights. Being permissive, it allows for derivative works to be released under different terms which can also be distributed commercially. Apart from Apache, Android is also released under this licence.

LGPL

Makes it possible to release derivative works under a different licence if it only makes use of LGPL's code as shared libraries. This allows for code to be used even in proprietary projects. This is why the licence is used primarily for software libraries. It is similar to GPLv3 in all other aspects.