A Proposed Addition to Open-Source Licensing for Improving Freedom of Use

Short paper

Juhani Naskali 0000-0002-7559-2595

Information Systems Science,
Turku School of Economics, University of Turku
Turku, Finland
juhani.naskali@utu.fi

Abstract. This paper identifies an often overlooked dimension of open source licensing, that of users being able to run distributed software freely and without discrimination, and suggests additions that would further secure freedom 0 to end users — the freedom for the user to use software however they wish. While, traditionally, open source licenses have left developers free to work with their code in any way they wish, and only secured rights for end-users through code modifications, new licenses rethink this paradigm. Furthermore, modern mobile application infrastructure is more restricted by design, and has changed the landscape of modified versions of applications (simultaneously widening the digital divide), and thus a new analysis is needed on how (and which) freedoms should be secured to ensure that free software remains free.

Keywords: open source, free software, FOSS, licensing terms, liberty, GNU GPL

1 Introduction

1.1 Free/Libre Open Source Software

Rights of ownership inherent in copyright can be granted to others using licensing. Open source (OS) software refers to licensing that allows licensees to freely distribute software code and redistribute it along with any changes. Open source is based on the idea of free software — an ideology of openness and inclusiveness, where required freedoms for using the software are secured for the public.

The freedom in free software refers to liberty, not cost, and this is often emphasized with the name FOSS or FLOSS — Free/Libre and Open Source Software. (GNU Project, 2001) Open source software is not necessarily free of cost (e.g. the application can be sold as a for-profit service while the code is open), and there are different licenses that grant differing amounts of freedoms to stakeholders. Developers are free to choose a license for their work that fits with their own values.

1.2 Freedom for everyone? Licenses of varying degrees of liberty

As with every kind of liberty in law, liberties secured through licensing come with opposing duties to others (Austin, 1885, p. 398). A license that protects (i.e. does not give away) patent rights of authors and contributors does so at the expense of end-users who now have to secure any patent licenses they require by other means. A copyleft or non-permissive open source license, one that requires contributors to license their derivative works using the same license, is using this duty of contributors to secure the right to use the software for the public. Both of these groups shape the license selection of OS projects Lerner & Tirole (2005).

Thus licensing liberty is a question of who holds what rights, and how this affects others. There is no ultimate universal liberty for everyone, as each liberty imposes some duties to others or signs away other rights. Practically, there are three somewhat overlapping parties to these license agreements: authors (original licencers), contributors (those who relicense/rerelease or sublicense their derivative works) and public (end-user licensees). The rights of these three parties are weighed against each other depending on what the license contains.

As choosing what rights to sign away with licensing is both a very personal choice and depends a lot on the type of project in question, it is a good thing that there is a variety of licenses to choose from: A for-profit endeavor can open their code to the public without losing rights to make a profit with their software; an ideological programmer can release code secure in the knowledge that the fruit of his labour will be free to all future end-users and not commercialized by others.

The degree of freedom and its recipients can be freely chosen, but while it is technically possible for everyone to write their own license, not everyone has the necessary legal skills to do so. It also takes a substantial amount of effort and skill to read through new licenses and understand their legal ramifications. A large enough diversity in licenses will create confusion, as it is difficult to grasp the totality of offered possibilities (Gomulkiewicz, 2009). It is clear that reusing shared, known licenses drives adoption.

1.3 The four user freedoms of free software

Free software ideology values the freedom of users. GNU GPL is founded on the four basic freedoms, which are an attempt to codify these values. The four basic freedoms are:

- the freedom to use the software for any purpose, (Freedom 0)
- the freedom to change the software to suit your needs, (Freedom 1)
- the freedom to share the software with your friends and neighbors, and (Freedom 2)
- the freedom to share the changes you make. (Freedom 3)

These four freedoms form the definition of free software, according to GNU Project (2001). Of particular interest to the topic of this paper is Freedom 0, which can be seen to be at the foundation of free software.

1.4 Researched dimensions of rights included in OS licenses

Lerner & Tirole (2005) conducted an empirical analysis on the determinants of license choice, and the scope of OS licenses on licensees freedom to use their software as they wish. They focused on two characteristics of OS licenses: copyleft, whether the full source code of any modified versions should also be published, and reciprocality, whether the source code can be mingled with source code utilizing different licensing terms.

These two main considerations, as well as author attribution, the inclusion informational texts (e.g. copyright and license notices) and DRM (Digital Rights Management) or patent usage rights in the license, are commonly used in mapping the different dimensions or aspects of rights granted by OS licenses (Schoettle, 2019; Hofmann et al., 2013).

One dimension not yet explicitly covered by previous examinations of the wide array of rights granted to licensees is the right to use the software as they wish (i.e. freely) without the need to alter source-code. Even though the importance of Freedom 0 has been discussed (GNU Project, 2001; Stallman, 2013) and included in the very definition of Free Software from the beginning (GNU, 1986), it has been mostly thought to be covered by the sharing of the source code and the right to modify and distribute it. After all, the idea goes, any unwanted limitations in the software can be changed or removed by licensees. But it can be argued that this way of securing freedom of usage is not strong enough to guarantee freedom 0 for everyone.

2 Missing dimension of securing user freedom to run software

2.1 User blocking is arguably allowed in open source software

While the four freedoms should secure the public the right to use software as they wish, there is no clear consensus about whether these freedoms are secured merely by sharing the source-code. An effective example of this can be seen in cases where distributed programs include hard-coded (embedded in the source code) functions that explicitly block users based on their affilitation (see Naskali, 2020). Affected users are unable to use the program without modifying the code and rebuilding the program. While Freedom 0 does not mean that application developers should be forced to add in functionalities for their users or that developers are in violation of GPL licensing if their programs fail to run with some inputs (GNU Project, 2001), it is something else entirely when a function that selectively identifies certain user groups and stops the program from running for them. This is clearly the case when the added function has no other purpose than blocking certain user groups and without it they would not be blocked (i.e. it is not an unintended consequence of something else or missing support for a protocol or service).

As censorship on the code level (e.g. domain blocking in source code) is a new phenomenon, it is difficult to say how prevalent these practices will become. Comments on implementations of domain blocking reveal that there is confusion on whether such an implementation of rights management is in compliance with OS licenses, free software philosophy or both/neither (e.g. Tusky, 2019). DRM has traditionally been seen to limit usage rights based on content copyright, not based on user affiliation. Nevertheless domain blocking is similar enough to traditional DRM, which is explicitly allowed in GPL as long as it is possible to remove such limitations by changing the program code, that it can be said to be in accordance of the licensing terms.

The technological industry has been moving towards permissive licenses, that allow for parts of the source code to be republished under closed licenses. Johnson (2021) data shows that while permissive licenses were used in 41% of open source licenses in 2012, in 2020 that number has risen to 76%. While it could be argued that as non-permissive licenses get less popular, the need for licenses that secure more freedoms for the end-users lessens, it does not necessarily follow. In fact, there is little downside to having multiple levels of freedom-securing licenses, especially when it comes to non-exclusive and discrete license dimensions (permissiveness and user blocking). There is perhaps some value in having clear options on this new dimension of user blocking.

2.2 Digital divide makes it impossible for general public to secure Freedom 0 using source code

The digital divide is defined as inequality between those who have access and means to use technology and those who do not. While these inequalities are not absolute, in that they can be bridged, nor the division clear-cut, it is a phenomenon that clearly exists and should be addressed (Van Dijk, 2006).

The effects of digital divide become more pronounced on second-and third-level digital divide, where the use and outcomes of differing skill-sets and possibilities are weighed (Scheerder et al., 2017). In the case of controlling usage of open source software, hard-coded control methods such as domain blocking and other types of censorship make it so that only those with access to computers and high-level programming skills have the freedom to use such software, and voice their opinions reliably (Naskali, 2020). This means that for any individuals not possessing the required programming skills to circumvent blocking functionality, Freedom 0 is not secured unless someone more skilled goes through the effort of removing said functionality and redistributing the unblocked version of the software.

If code is introduced in an open source project with the goal of blocking certain groups from making use of said software, it is not necessarily enough that the code can be copied and changed to remove the changes. If it is indeed an important goal of (some of) the developers to consciously block such use, it is possible that the blocking code will be obfuscated and made difficult to remove, or that other types of difficult-to-remove DRM will be introduced. In the current era of centralized platforms, it is even possible to release open-source software that depends on a centralized server network that doesn't allow forks on it. Both these methods would further exasperate the digital divide, and make securing freedom 0 difficult or impossible even for experienced programmers.

Much of open source ideology hinges on the fact that development is done according to good programming practices and technology is not influenced by outside forces such as politics. This is perhaps not quite as true as it was a a couple of decades ago. In a more uncertain environment, it is completely conceivable that a developer might want the option to protect the freedom of the public to use their software freely, and licensing is the correct tool for guaranteeing such freedoms.

Digital divide makes it imperative for IT specialists to pay closer attention to public liberties, as the majority of users do not have the requisite skillset to modify programming code, rebuild it to new binaries and distribute their fork of the software. This majority is silent due to not being represented on the discussion forums of code development platforms. Moreover, the walled gardens of modern mobile platforms make it even more complicated to distribute your own version of an applications.

2.3 New ethical source licenses

In past years, there has been more interest in open source licenses based on ethical values (Goodman-Wilson, 2020). For example, Ethical Source tries to drive licenses that prevent software use for harm[1] and their GitHub repository's[2] first commit was made in 2019.

Notably, such licenses cannot be considered open source, as restricting the use of software by licensing is against the open-source ideology and Open Source Initiatives rules for open source licenses (2019), which state that open-source licenses must comply with ”No Discrimination Against Persons or Groups” and ”No Discrimination Against Fields of Endeavor”. Such license is not open. This new type of licensing nevertheless serves as an example of more ethical consideration being put in licensing, as well as the cultural drive for more restrictive measures in licensing — more restrictive for the public, securing more rights for the licensers (i.e. limiting what others do).

2.4 Examination of GNU GPLv3

GNU GPL is founded on the four basic freedoms listed in the introduction. The license has been formulated to secure these freedoms through the sharing of source code.

[1] https://ethicalsource.dev/licenses/

[2] https://github.com/EthicalSource/ethicalsource.dev

Distribution agreements have been consciously left outside this license, presumably because user freedom is secured through the availability of the code, even in the case of closed or paid binary distribution.

GPLv2 was updated to GPLv3 in 2007 to include new freedoms that were previously not secured for end-users, and includes protections against ”Tivoization”, that prevent users from freely using the software by using hardware-based limits to code modification; and laws prohibiting free software, that prevent using DRM laws to prevent software distribution and discriminatory patent deals. In other words, GPL-licensed software can include DRM as long as its removal is not made impossible with special hardware or laws.

GPLv3 (GNU, 2007) is used as an example in this paper because it is explicitly founded on the four basic (user) freedoms and is one of the most freedom-securing and well-known OS licenses. Nevertheless, the proposed changes could be built on top of any number of licenses or formulated as a new license with a completely different name.

GPL and open-source licenses currently do not include protections for users to run the distributed binaries without impediments. It must be understood that this is not necessarily a bad thing. Not securing these liberties ensures more freedom to the licensers to do as they wish (e.g. implement DRM), and this can be a good thing.

I argue that the following changes are in the spirit of GNU’s four basic freedoms, but should not necessarily be at the core of all GPL licenses. In a similar fashion as LGPL has a slightly different emphasis on the open source ideals compared to GPLv3, these proposed additions would constitute another take on how the license weighs liberties between authors, contributors and the public, granting some freedom away from authors and contributors in favor of the public. I’m not arguing that this should happen instead of what e.g. GPLv3 does — I’m only arguing it should be an explicitly stated option.

3 Proposed additions, by example of GPLv3

First proposed addition to GNU GPLv3 Chapter 5. Conveying Modified Source Versions.

e) The work must not include any methods or additional protections for restricting the use of the work by particular persons, groups or fields of endeavor.

This addition would disallow the use of DRM (digital rights management) that targets persons or groups. Censorship systems such as domain blocking and network-based DRM would also fall under this category.

Second proposed addition to GNU GPLv3 chapter 6. Conveying Non-Source Forms

1. *Any object code conveyed must not include extra blocking methods aimed at discriminating agains persons, groups or fields of endeavor specified in Chapter 5.*

The object of this addition would be to secure Freedom 0 when using conveyed non-source forms (i.e. binaries). This would (most importantly) include mobile apps downloaded from official app stores.

This addition might be redundant with the previous addition blocking the use of said blocking methods in the source code. It is included here to facilitate discussion, as the main part of the addressed problem is caused by distributed binaries, especially in the walled-gardens of mobile app stores. Perhaps it would be best to only target the source code and not widen the license to touch this part of distribution. Or perhaps it would be better to keep total freedom for developers when it comes to the source code, and only include this addition or allow distribution as long as there is also a version available without restrictions.

4 Thoughts

4.1 Would a license with the proposed changes still be open source?

The proposals would sign away more rights from contributors, should they wish to restrict the right of usage by the public. It could be argued that such licensing is no longer open source, or that it is less open, because use of the source code is restricted in some manner (especially in the first proposal). But the restrictions are comparable to the duties of needing to provide the source-code along with distributed binaries or not being allowed to relicense the work and choose their own license — the contributors are given duties that correspond to ensuring the freedoms of the public.

Perhaps such a license would not be open source, but rather part of the ethical licenses family. Nevertheless, such a distinction is somewhat semantic in nature. The cure issue is whether more weight is put on the rights of the developers (both original licencers and new licensees) or the public (end-users, whether developers or not).

4.2 Proposed changes are arguably in line with GNU values

GNU GPL is based on the four basic freedoms of users, and as articulated by Stallman (2013), freedom 0 ”means that the distribution of the software does not restrict how you use it”. Stallman’s example is vivid: ”Imagine selling pens with conditions about what you can write with them; that would be noisome, and we should not stand for it. Likewise for general software.”

The digital divide mentioned in chapter 2.2 makes it imperative that user freedom be guaranteed, moreso when development and distribution options get more difficult. Mobile application distributions systems are perhaps a special case when it comes to software distribution, as they mostly rely on ”walled gardens”, a centralized distribution system in the form of app stores[3]. Installing your own version of an application requires at the very least a development environment dedicated to the chosen platform and a high level of technical know-how; possibly also payment to the required distribution platform, if one wants to share the application or install it without resorting to developer options. In such a case, Freedom 0, the freedom to use the software for any purpose, is not secured for all users without additions such as the ones proposed.

In comparison, the proposed changes would be clearly in conflict with the values of some other ethics-based licenses that explicitly restrict the use of software, e.g. the

[3] While Android allows the installation of apk files outside Play Store, the option is off by default and can be difficult to do without technical skills.

NoHarm license[4] that forbids use related to things such as slavery, gambling, tobacco, hate speech or discrimination. This is not a problem, as a wider selection of licenses ensures that people can choose how their software is shared and used. The proposed additions would not affect those who choose not to use a license that includes them.

Arguably, the proposed changes would go well together with a new version of GNU GPL. Just as GPLv2 was updated to GPLv3 to protect user freedom from tivoization, laws prohibiting free software and discriminatory patent deals, these new additions would further protect users from new discriminatory restriction methods. As such, they should not be a replacement license but a complementary one or a completely new license.

4.3 Comparisons with similar end-user freedoms not included in GPLv3

Money for nothing

GNU GPL explicitly allows selling copies of free software[5]. This provides a good example of how 'free' in free software refers to liberty (of code) and not cost (of distribution). The proposed changes do limit freedom to work with the code, which could bear further examination to ensure that this does not create unforeseen problems.

Limiting usage through DRM

Perhaps the closest thing to the proposed changes is the use of DRM, which is explicitly allowed under GPLv3 licensing[6]. But even DRM is disallowed when protected by tivoization or law, and a case could be made that similar protection is provided by modern mobile app distribution platforms, which should be included in similar fashion.

Voters are not allowed to change open source code in voting machines

The GPL FAQ gives an example of voting machine users not being allowed to change the code running on voting machines. Such abilities to control running software are not required by GPLv3 licensing.

The FAQ states that such liberty is not required, as the voter doesn't get posession of the object code, even temporarily, and so the voter also doesn't get possession of the binary software. The FAQ also states that ”voting is a very special case”, so the applicability of these thoughts on software in general and distribution of mobile applications specifically is arguable.

4.4 Possible problems

The proposed changes should be scrutinized by the open-source community and, specifically, legal experts, to make sure that their scope is as-intended. Without sufficient legal expertise, it is possible that there are large unintended consequences, and these should be found out and eliminated before releasing a full license.

[4] https://github.com/raisely/NoHarm/

[5] https://www.gnu.org/philosophy/selling.html

[6] https://www.gnu.org/licenses/gpl-faq.htmlDRMProhibited

The proposed changes do limit the freedoms of contributors significantly. It might be possible to limit these freedoms less, while still protecting the public freedom of use, by e.g. requiring the conveying of non-source forms without the listed protection methods. This would allow contributors to release code and binaries with discriminating restriction methods as long as they also release versions without restricting methods.

Arguably, a more clear-cut division of licenses makes more sense: if you wish to restrict the use of your software by code, choose a license that clearly allows such restrictions; if you wish for your program to be open and freely usable, choose a license that disallows restricting methods.

5 Conclusion

This paper identified user freedom to run a software as they wish as a dimension of open source licensing that has so far been left undefined in commonly used OS licenses. This new dimension should perhaps be considered in scientific literature along with the more common dimensions of copyleft/permissiveness and the securing of patent rights.

In practice, while there are already many licenses to choose from, and there is nothing inherently wrong in the current formulation of GPLv3, there is some confusion on whether blocking particular persons or groups is compliant with it (Naskali, 2020). When some people interpret the licensing terms in a way that it already forbids things like hard-coded domain blocking of user groups, and others view this type of application behaviour as license compliant, it seems like some clarification is in order.

Functionality overriding Freedom 0 should be clearly allowed or clearly disallowed in OS licenses, in order for licensers to make informed choices on the matter. If the question is made explicit in the form of new licenses with the additions proposed in this paper, licensees could choose between licenses based on their own wishes, and contributors using freedom 0 securing licenses could be secure in the knowledge that the project they are working on will stay free.

References

Austin, J. (1885). *Lectures on jurisprudence, or, The philosophy of positive law*. John Murray.

GNU (1986). Gnu's bulletin volume 1 no.1.
Retrieved from `https://www.gnu.org/bulletins/bull1.txt`

GNU (2007). Gnu general public license v3.
Retrieved from `https://www.gnu.org/licenses/gpl-3.0.html`

GNU Project (2001). What is free software?
Retrieved from `https://www.gnu.org/philosophy/free-sw.html`

Gomulkiewicz, R. W. (2009). Open source license proliferation: Helpful diversity or hopeless confusion. *Wash. UJL & Pol'y*, *30*, 261.

Goodman-Wilson, D. (2020). Open source is broken. FOSDEM. Brussels, Belgium.
Retrieved from `https://mirrors.dotsrc.org/fosdem/2020/UB5.230/ethicsoss.webm`

Hofmann, G., Riehle, D., Kolassa, C., & Mauerer, W. (2013). A dual model of open source license growth. In *IFIP International Conference on Open Source Systems*, (pp. 245–256). Springer.

Johnson, P. (2021). Open source licenses in 2021: Trends and predictions. Accessed 2021-06-07.
Retrieved from `https://www.whitesourcesoftware.com/resources/blog/open-source-licenses-trends-and-predictions`

Lerner, J., & Tirole, J. (2005). The scope of open source licensing. *Journal of Law, Economics, and Organization*, *21*(1), 20–56.

Naskali, J. (2020). Hard-coded censorship in open source mastodon clients—how free is open source? *Proceedings of the Conference on Technology Ethics 2020*.

Scheerder, A., van Deursen, A., & van Dijk, J. (2017). Determinants of internet skills, uses and outcomes. a systematic review of the second-and third-level digital divide. *Telematics and informatics*, *34*(8), 1607–1624.

Schoettle, H. (2019). Open source license compliance-why and how? *Computer*, *52*(8), 63–67.

Stallman, R. (2013). Why programs must not limit the freedom to run them.
Retrieved from `https://www.gnu.org/philosophy/programs-must-not-limit-freedom-to-run.html`

Tusky (2019). Tusky pull request #1303 · tuskyapp/tusky.
Retrieved from `https://github.com/tuskyapp/Tusky/pull/1303/`

Van Dijk, J. A. (2006). Digital divide research, achievements and shortcomings. *Poetics*, *34*(4-5), 221–235.