

SCRIPT-ed

Volume 3, Issue 2, June 2006

GNU General Public License v3: A Legal Analysis

*Andrés Guadamuz González**

Abstract

This paper offers a first-look legal analysis of the draft version 3 of the GNU General Public License, and will also look at the debate that it has generated in the Free and Open Source community. The paper will answer the following vital questions: Is the GPL v3 different in any fundamental way from GPL v2? What are the main differences? What will be the impact of the new GPL to the problem of software patents? Is there an incompatibility problem with previous versions of the licence? How does it compare with other existing licences?

DOI: 10.2966/scrip.030206.154

© Andres Guadamuz 2006. This work is licensed through a Creative Commons Attribution-NonCommercial-NoDerivs 2.5 Scotland License.

* Lecturer in Law, University of Edinburgh. This paper was presented at the Open Source systems 2006 Conference in Como, Italy (June 2006). Thanks to David Alexander Russell for pointing out an error in the paper.

1. Introduction

The draft of the version 3 of the GNU General Public License (GPL) was released in January 2006, causing quite a stir in the Free Software and Open Source (hereby referred to as FOSS) communities. The draft is still under discussion at the time of writing, but it is becoming clear that the new GPL could split the non-proprietary community by making developers choose between different types of development ideologies. The purpose of the present paper will be to provide a legal analysis of the clauses contained in the new draft in order to answer whether the existing GPL requires an update.

2. GPL basics

It is not an exaggeration to state that the GNU General Public License (GPL) is the most important licence in the FOSS movement. At the time of writing, 67% of all projects listed in the SourceForge open source repository are released under the GPL.¹ But why is it so important? What makes the GPL the licence of choice for non-proprietary development?

The GPL was first drafted in 1985² by Richard Stallman of the Free Software Foundation (FSF) in order to accommodate the ideas of free distribution of source code espoused by their development philosophy.³ The licence was designed to protect software released by the FSF with something that Stallman called copyleft.⁴ This concept was to allow the further distribution and modification of software released under its terms, but it would not allow proprietary developers to come and “close” the released software.⁵ The version 2 (and current) version of the licence was released in 1991, and it was drafted by Stallman and by law professor Eben Moglen.⁶

The GPL can be described as a form contract⁷ that ensures that the source code is passed on, but anyone who redistributes it, with or without changes, must pass along the freedom to further copy and change it. This places a burden on the person transferring the software; the burden is that the software must remain “free”, as

¹ Out of 73,978 projects, 50,013 were released with the GPL. See: <http://sourceforge.net/softwaremap>.

² Although the official dating of version 1 is 1989. This is because the GPL was known originally as the EMACS General Public License.

³ Stallman R, “The GNU Operating System and the Free Software Movement”, In DiBona C, Ockman S and Stone M, *Opensources: Voices from the Open Source Revolution*, London: O'Reilly (1999), p.59.

⁴ For more about the concepts of copyleft, see: Guadamuz A, “Viral contracts or unenforceable documents? Contractual validity of copyleft licenses”, 26(8) *European Intellectual Property Review* 331-339 (2004).

⁵ Moody G, *Rebel Code: Linux and the Open Source Revolution*, London: Penguin (2002), pp.26-29.

⁶ Ibid.

⁷ Some people do not believe that the GPL is a contract. See: Jones P, “The GPL Is a License, not a Contract”, *Linux Weekly News*, December 3 (2003). For an excellent rebuttal of this argument, see: Rosen LE, *Open Source Licensing: Software Freedom and Intellectual Property Law*, Upper Saddle River, N.J.: Prentice Hall PTR (2004), pp.59-66.

defined by the FSF and the GPL.⁸ This is different from just placing software in the public domain because the person making use of the free code can subsequently copyright it and release it in proprietary format.⁹

The GPL is the legal framework that sustains most of the software copyleft system.¹⁰ It reads as a mixture of a legal contract and an ideological manifesto. The preamble to the work restates clearly some of the most common beliefs of free software and the non-proprietary approach, with several admonitions about the meaning of the word “free”. The main point is that, as mentioned before, the source code must be made available to the users. The preamble states:

“For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.”¹¹

The licence specifies that this is achieved by two means: by protecting the software by means of copyright; and by providing the users with a licence that gives them the freedom to use and modify the software in any way they see fit if they meet the stated conditions. The main body of the licence reiterates these ideas. Section 1 for example states:

“1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.”¹²

This section also mentions that the user can make monetary charges when distributing the copy as long as the charges are for expenses in making copies of the software, which is also consistent with the general free software characteristic that does not discriminate against commercial software as long as it is not proprietary¹³ commercial software.

Many of the provisions of the GPL can be found in other non-proprietary software licences. What makes the GPL unique is section 2(b) of the licence, as this is where the restrictions against using the software to create commercial software are specified. The section reads:

⁸ Or as they like to repeat, free as in freedom, not as in beer.

⁹ Lambert P. “Copyleft, copyright and software IPRs: is contract still king?” 23(4) *European Intellectual Property Review*, (2001), pp.165-171.

¹⁰ The full text of version 2 of the licence can be found at: <<http://www.fsf.org/licenses/gpl.html>>

¹¹ Free Software Foundation. *GNU General Public License v2*, Preamble.

¹² *Ibid*, s.1.

¹³ Proprietary is generally equated to “closed” source. See: O'Sullivan M, “Making Copyright Ambidextrous: An Expose of Copyleft”, *The Journal of Information, Law and Technology* (JILT) 2002 (3), @ <http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2002_3/osullivan/>.

“2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: [...] b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.”¹⁴

What this means is that any software developed by using the open source code of the copyleft program must not charge for the derivative product, and most importantly, must ensure that the GPL is transferred to further users of the derivative software. This type of licence has been aptly named a “viral contract” by Professor Radin, defining them as “contracts whose obligations purport to ‘run’ to successor of immediate parties”.¹⁵ These contracts would then spread in a viral form, as the licensee must include the terms of the GPL in any subsequent licence they will include to their derivative work because that obligation is part of the contract, and then those subsequent licensees will have to impose the same contractual terms in further licences that they perform, *ad perpetuam*.

Despite the fact that copyleft licences tend to promote the free software principles and the definitions drafted by the FSF and Stallman, it must be pointed out that some of the contractual restrictions existing in licences such as the GPL have prompted some criticism from enterprises and commercial users.¹⁶ Despite these criticisms and an incessant campaign from some proprietary software developers against it, the GPL has stood well against most challenges, as exemplified by its longevity and the relatively small litigation that it has generated.¹⁷

3. GPL v3

While the longevity and success of the GPL have been a good testament as to its legal validity,¹⁸ it was felt at the FSF that it was time for an updated version to fine-tune some legal issues and to respond to developments in the Free Software community.

¹⁴ Ibid, s.2(b).

¹⁵ Radin, M. “Humans, Computers, and Binding Commitment”, *Indiana Law Journal*, Vol.75, Fall 2000, p.1125. Some people in the Free Software community do not like the term “viral”, but it has become more prevalent in the literature. For example, see: Wichman T, “Firms’ Open Source Activities: Motivations and Policy Implications”, Report for the *Free/Libre Open Source Software: Survey and Study* (2002). @: <<http://strategicadvice.net/floss.pdf>>.

¹⁶ For some criticisms, see: Guadamuz A, “Legal Challenges to Open Source Licences”, 2(2) *SCRIPT-ed* 301-308 (2005).

¹⁷ See: Miller JS, “Allchin’s Folly: Exploding some myths about open source software”, 20 *Cardozo Arts & Entertainment Law Journal* 491 (2002).

¹⁸ For further legal analysis of its validity, see: Gomulkiewicz R, “De-Bugging Open Source Software Licensing”, 64 *University of Pittsburgh Law Review* 75 (2002).

Prior to the release of the draft, Stallman and Moglen identified key issues that they believed needed to be addressed by the licence.¹⁹ These were:

1. Internationalise the licence: While GPL v2 was drafted by with American law in mind, it is purported to adhere to the international copyright principles present in the Berne Copyright Convention.²⁰ While the only ruling in favour of the GPL has taken place outside of the United States,²¹ it has been felt that it requires tightening of its international compliance.
2. Standard setting: Stallman and Moglen argue that the GPL has become an international standard for the FOSS industry, so any update has to take this into consideration.
3. Responding to changing circumstances: The FOSS community, and in particular the principles of Free Software, are faced with new threats, such as patentability of software, trusted computing and digital rights management. The GPL must change to answer those threats.²²

So, what is contained in the new licence? The first draft of the version 3 of the GPL is longer, and unfortunately less clear than version 2.²³ This is a real problem, as it has been often pointed out that the existing wording of the GPL is not the clearest example of copyright licensing, its interpretation and broad generalisations are the subject of endless legal interpretation.²⁴

The aforementioned document stating the need for a redraft prompted expectations that the new draft text would attempt to overhaul the protection of GPL software against software patents and it would perhaps update the copyleft principle present in the existing section 2(b). Although the actual draft mentions these topics, the real surprise came with regards to digital rights management (DRM), in particular technical protection measures (TPM). The new GPL could have a lot of consequences for the future of open source usage in the entertainment industry. One of the most contentious sections is section 3 on DRM:

“Regardless of any other provision of this License, no permission is given to distribute covered works that illegally invade users' privacy, nor for modes of distribution that deny users that run covered works the full exercise of the legal rights granted by this License.”²⁵

In other words, the distribution of derivatives with works that contain certain restrictive types of DRM will be prohibited. While it must be commended that this

¹⁹ Moglen E; and Stallman R, *GPL Version 3: Background to Adoption*, June 5 (2005), @: <<http://www.fsf.org/news/gpl3.html>>.

²⁰ Berne Convention for the Protection of Literary and Artistic Works 1886.

²¹ Höppner J, "The GPL Prevails: An Analysis of the First-Ever Court Decision on the Validity and Effectivity of the GPL", 1(4) *SCRIPT-ed* 662 (2004) <<http://www.law.ed.ac.uk/ahrb/script-ed/issue4/GPL-case.asp>>.

²² Stallman and Moglen, supra note 19.

²³ The draft can be found here: <<http://gplv3.fsf.org/draft>>.

²⁴ Gomulkiewicz R, "General Public License 3.0: Hacking the Free Software Movement's Constitution", 42(4) *Houston Law Review* 1015 (2005), pp. 1034-1036.

²⁵ s3 of GPL v3.

clause is very specific against specific types of technical protection measures, the provisions are still hamfisted when read in conjunction with the next paragraph.

“No covered work constitutes part of an effective technological protection measure: that is to say, distribution of a covered work as part of a system to generate or access certain data constitutes general permission at least for development, distribution and use, under this License, of other software capable of accessing the same data.”²⁶

This paragraph specifically excludes all works distributed under the GPL from the anti-circumvention measures in the WIPO Copyright Treaty²⁷ (WCT) by specifically stating that the licensed software shall not constitute "an effective technological protection measure", and it would therefore not apply for such protections. This seems a much more effective way of dealing with TPMs than the broader and more difficult to implement version of the first paragraph.

Surprisingly, the new version is not as adamant against software patents as it was expected as a reflection that some of the biggest open source and free software players in the United States are acquiring patents as well.²⁸ Stallman and Moglen seem very pragmatic regarding software patents, and recognise that FOSS developers may be involved in complex patent licensing transactions, therefore their implicit recognition of the status quo.²⁹

The new draft expands on the implicit patent licensing of the existing GPL and make it an explicit patent grant in section 11 of the draft. The licence prohibits distribution of the software by people who have initiated or brought patent suits arising from the software. This is a clever attack on software patents, it does not forbid people to apply for them, but it frowns upon enforcement. Strangely, the licence allows developers to add a stronger patent retaliation clause that is compatible with the licence. This clause is spelled out, but it is not really a part of the licence, it is just compatible with it. The compatible patent clause states:

“e) They may impose software patent retaliation, which means permission for use of your added parts terminates or may be terminated, wholly or partially, under stated conditions, for users closely related to any party that has filed a software patent lawsuit (i.e., a lawsuit alleging that some software infringes a patent). The conditions must limit retaliation to a subset of these two cases:

1. Lawsuits that lack the justification of retaliating against other software patent lawsuits that lack such justification.

²⁶ Ibid.

²⁷ Specifically, Art. 1 of the WCT, which states that “Contracting Parties shall provide adequate legal protection and effective legal remedies against the circumvention of effective technological measures...”

²⁸ Apache, IBM and Novell have applied for thousands of software patents in the United States. For more about this, see: Guadamuz A, "The Software Patent Debate", 1(3) *Journal of Intellectual Property Law & Practice* 196-206 (2006).

²⁹ See their draft rationale here: <<http://gplv3.fsf.org/rationale>>.

2. *Lawsuits that target part of this work, or other code that was elsewhere released together with the parts you added, the whole being under the terms used here for those parts.*"³⁰

This may prove to be terribly confusing for the average reader in my opinion, and a bad drafting practice.

Another interesting feature of the new draft is that it revamps the old copyleft clause 2(b). As explained earlier, under the current GPL the copyleft aspects only apply to derivative works that are distributed to the public. In other words, you take a work under the GPL, change it and then distribute your own adaptation. Then you have to re-distribute it under the GPL. This simple rule would generally provide clear-cut cases in which the GPL would apply and where it would not. For example, imagine that a hardware developer creates a driver module that interacts with the Linux kernel (distributed under the GPL). It is not part of the kernel, but it interacts with it. Under GPL v2, it is clear that this module is not a derivative, and therefore it does not need to be distributed under the GPL.

What happens with GPL version 3? The situation with derivatives is less straightforward. The copyleft clause in the new GPL has been given a tremendous boost. Users and developers still have the right to install and use GPL software without restrictions. Developers also have the right to privately modify the software, unless they have initiated a patent suit "against anyone for making, using or distributing their own works based on the Program." The problem is that of modifications that are distributed. Consider section 5(b) (the old section 2.b):

*"b) You must license the entire modified work, as a whole, under this License to anyone who comes into possession of a copy. This License must apply, unmodified except as permitted by section 7 below, to the whole of the work. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it."*³¹ (Emphasis mine).

This would certainly apply to anyone who includes any sort of modification of code into their work. Imagine for example that you include modified modules from the Linux kernel in your work in order to provide compatibility with the kernel. It seems like you would definitely have to licence the entire program under the GPL! This would not apply if the work can be clearly identified as not being a derivative. The new GPL says:

"These requirements apply to the modified work as a whole. If identifiable sections of that work, added by you, are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works for use not in combination with the Program. But when you distribute the same sections for use in combination with covered works, no matter in what form such combination occurs, the whole of the combination must be

³⁰ S7(e) draft GPL v3.

³¹ S5(b) draft GPL v3.

licensed under this License, whose permissions for other licensees extend to the entire whole, and thus to every part of the whole."³²

This seems like overboard protection. Even if a work is not based on GPL software but it is distributed in combination with covered works, then it must be licensed under the GPL. What will happen with open source projects that do not choose to use the GPL, such as Apache? And what about interfaces? At first glance, it seems that this would also affect drivers and modules that interact with the kernel. However, the draft further confuses this by stating:

*"A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. Mere inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate."*³³

This paragraph seems to generate more legal uncertainty, something that has not been lost on commentators of the draft.³⁴ This paragraph tries to rationalise specific cases in which the revamped GPL copyleft clause will apply by inventing a new definition to what a compilation is. They try to determine that the software distributed in the same distribution medium has to be GPL if it is a "compilation", but not if it is an "aggregate". Why create the new terminology?

It seems like the drafters of the new GPL realised that the new copyleft clause would be controversial when they included an excuse within it:

*"Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program."*³⁵

Why excuse the clause in the licence if there is nothing to excuse?

4. The verdict so far

It is difficult to ascertain yet what will be the final result of the draft GPL v3. Stallman, Moglen and the FSF must be commended by the inclusive and open process with which they have undertaken the drafting process. The GPL v3 has been open for discussion to the public since January 2006, a process that will continue with specific working groups (both open and closed to public participation), and a further draft can be expected soon. By reading some of the comments, it seems that the FOSS community has been finding specific areas of special concern, particularly the

³² S5 draft GPL v3.

³³ Ibid.

³⁴ For a series of comments, see: <<http://gplv3.fsf.org/comments/>>.

³⁵ S5 draft GPL v3.

revamped copyleft clause and the DRM provisions. Take for example a comment to section 4 with regards to DRM and software that violates privacy:

“Frankly, this clause seems off-topic. Why not, for instance, deny distribution to covered works that are components of weapons of mass destruction? The whole issue of what constitutes privacy is something of an open issue these days, when everybody submits their corporate shopping ID at the grocery store checkout counter and thinks nothing of having their most everyday habits and activities tracked, profiled, and distributed to anyone willing to pay for them.”³⁶

The DRM issue has become a real sticking point with the wider developer community. The most controversial indictment of the draft, and perhaps the most damaging, was delivered by Linus Torvalds, the father of the Linux kernel. Torvalds has declared that the Linux kernel will not be released using GPL v3, but that it will continue to use version 2. His problem lies specifically with the strong prohibition on the use of DRM found in section 3. Torvalds commented that the new GPL would require people to give control of their private keys, which he will not do. He stated that:

“Sure, DRM may mean that you can not `_install_` or `_run_` your changes on somebody else's hardware. But it in no way changes the fact that you got all the source code, and you can make changes (and use their changes) to it. That requirement has always been there, even with plain GPLv2. You have the source. The difference? The hardware may only run signed kernels. The fact that the hardware is closed is a `_hardware_` license issue. Not a software license issue. I'd suggest you take it up with your hardware vendor, and quite possibly just decide to not buy the hardware. Vote with your feet. Join the OpenCores groups. Make your own FPGA's.”³⁷

This should come as no surprise to those who follow open source debates. Back in 2003 Torvalds had already expressed that he thought some DRMs were useful, as there seems to be a distinction between the more general DRM and the more restrictive TPM.³⁸ There are already open source projects that intend to allow artist control,³⁹ while at the same time provide users the tools to use general DRM to monitor and track their works, but not to restrict access, which is what a technical protection is. DRM is a very broad term, and licence management can be considered in that light.

To make matters worse, Stallman has mentioned that Torvalds has no say in what happens to the Linux kernel, and that the kernel developers are the ones who will

³⁶ Comment by user ‘kop’ in the Comments section, see: <<http://gplv3.fsf.org/comments/>>.

³⁷ Linus Torvalds, quoted in: Barr J, “Torvalds versus GPLv3 DRM restrictions”, *NewsForge*, February 2 (2006), @: <<http://trends.newsforge.com/article.pl?sid=06/02/02/1636216>>.

³⁸ Bechtold S, "The Present and Future of Digital Rights Management: Musings on Emerging Legal Problems", Becker E; Buhse W; Günnewig D; and Rump N (eds.), *Digital Rights Management: Technological, Economic, Legal and Political Aspects*, Springer, Berlin (2003), pp. 597-654.

³⁹ For example, Authena, see: <<http://authena.org/>>.

decide over the licence.⁴⁰ This could prove tricky, Torvalds owns the trade mark to Linux, and he is one of the co-authors of the Linux kernel and therefore not the sole proprietor, so he may not be able to make unilateral decisions about licensing.

The problems about licensing only serve to stress the controversial nature of the new draft. As the GPL has become the *de facto* constitution of the FOSS community, any change to that constitution will generate considerable problems.

5. Conclusion

The final verdict on the draft will undoubtedly come soon, but it seems clear that the draft in its present form may have some problems, and will require further adjustment to make it as widely accepted as GPL v2.

It is clear that the new version of the licence has lost clarity and gained length. This would not be a problem as such if this was followed by a strengthening of the existing provisions, but this does not seem to be the case at the moment. While there are some elegant and clever clauses in the new draft, others seem to be filled with explanatory paragraphs that seem to be out of place in a legal document. The draft could pick some drafting tips from the smaller and more elegant licences that solve some of its issues in a more efficient manner.

⁴⁰ Vaughan-Nichols S, "Torvalds hasn't ruled out GPL 3 for Linux", Linux-Watch, February 7 (2006), @: <<http://www.linux-watch.com/news/NS3301105877.html>>