A framework for understanding the impact of GPL copylefting vs. non copylefting licenses

by Philippe Aigrain

Views presented in this paper are only the author's and do not necessarily reflect the official European Commission's view on the subject.

Abstract: This text provides a framework to discuss the consequences of licensing choices that are applied to publicly funded libre software. It discusses these choices from the angle of general public interest and policy. It concludes that one is led to prefer GPL copylefting licensing especially for any libre software component whose development is publicly funded, and when it is providing functionality that plays or may later play a critical role as part of the common infrastructure of the information society. At the same time, some interesting questions derive from a better understanding of the reasons that lead some to disagree with the choice of copylefting.

The debate on whether libre software should be released under copylefting licenses (in practice the GNU General Public license) or under non-copylefting licenses such as the BSD or Apache-style licenses is a constant in the history of libre software. One would even hesitate to trigger again the flow of flaming messages that unvariably follow any statement on this subject. However, two more recent factors make it absolutely necessary to establish new foundations for this debate. The first factor resides in the attacks from some major proprietary software vendors against the GPL. These attacks were first expressed in a very agressive manner, which backfired and brought the whole libre software community together in support of the GPL. They have since continued in more subtle ways, within standardisation arena, through lobbying and in PR campaigns. The second factor is that, as public support of libre software gains momentum, the debate on copylefting vs. non-copylefting has recently arisen in specific terms for publicly funded software development.

Leaving aside the agendas of proprietary vendors, it is clear that many of the people who disagree on this issue share common values, namely freedom, cooperation and the passion to create useful things. If they keep disagreeing despite having these values in common, it must be because they look at the issue from different angles. It is thus useful to provide a framework that explains what these different angles are, and sheds some light on the consequences of going one direction or the other.

Let us consider a software component to be released as libre software. Imagine someone who considers only the immediate consequences of the choice of license, and that his/her vision has a scope similar to the black rectangle on the top-left part of the figure 1. To that person, the choice of the copylefting license is clearly a restriction, since it will close some possible dissemination routes for the software. For instance, it will not stop it from being used as an underlying platform on top of which proprietary software is run, or it will not stop its usage on top of proprietary software, but it will forbid its exploitation within proprietary software. In contrast, a non-copylefting license will seem to open more dissemination routes, specially if despite being non-copylefting, it is compatible with the GPL. In our graphical representation, the dissemination paths corresponding to the blue arrow will be made impossible by the GPL. This view is not an illusion, it is clear that the immediate usage might in many cases be less widespread. Anyone adopting a narrow vision, focussed on the immediate and isolated consequences of the license choice might prefer non-copylefting licenses from the very points of view of freedom, enabling maximum co-operation, and providing as many people as possible with a useful piece of software.

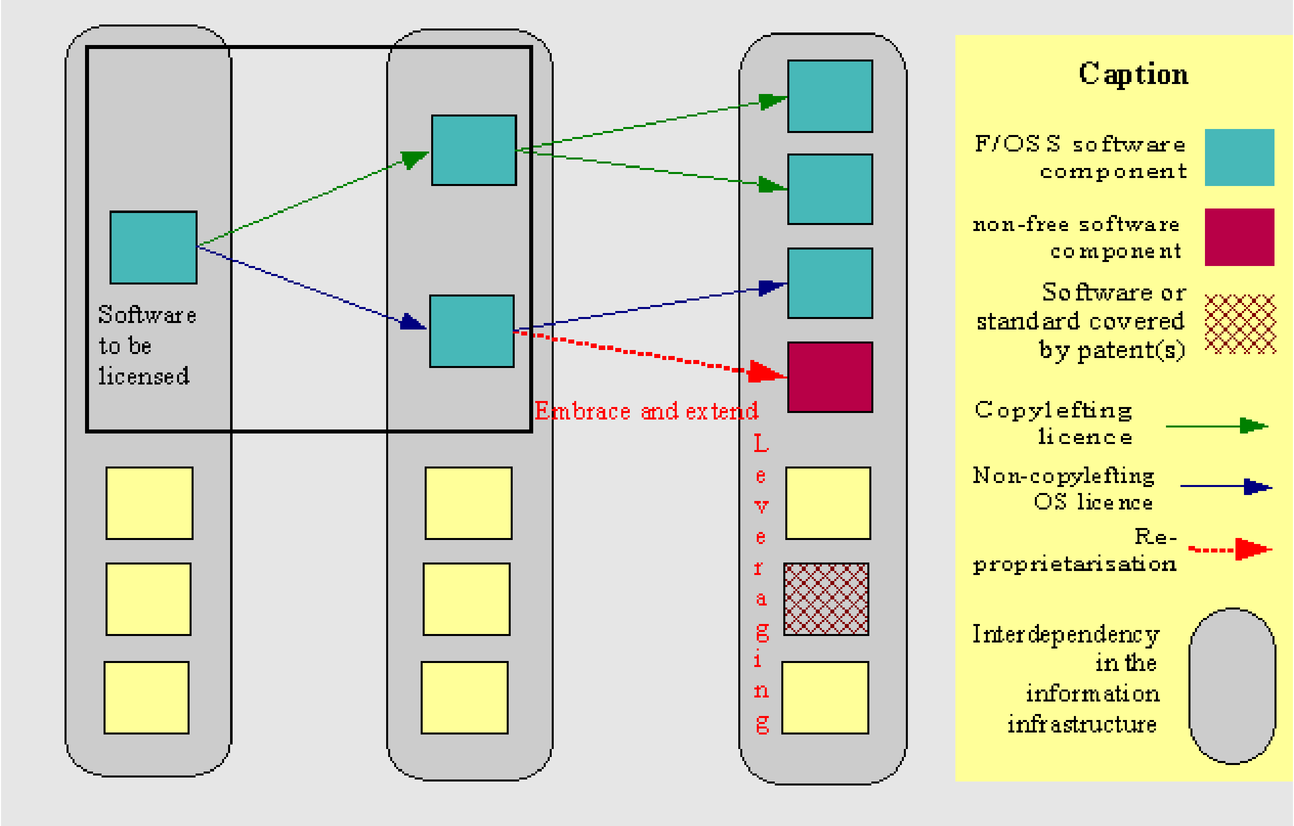
Software to be licensed
Embrace and extend
Leveraging
Caption
F/OSS software component
non-free software component
Software or standard covered by patent(s)
Copylefting licence
Non-copylefting OS licence
Re-proprietarisation
Interdependency in the information infrastructure

Figure 1. Graphical representation of the framework

Such an analysis is severely shortsighted in some essential cases, for 2 critical reasons.

First, software components are not used in isolation: any real-life activity uses many layers of software, and complex chains of components. Creating some media contents and distributing them will mobilize software for acquiring, editing, describing, coding and compressing them, making them available on servers, distributing them on networks, accessing them on clients, etc. Each of these software will depend on underlying hardware and software platforms. Developing a Web-based electronic booking service for a bed and breakfast will also require multiple functional components, each dependent on many layers of the common infrastructure. The main property of the software infrastructure is interdependency.

Second, the software of today will change tomorrow, as will its terms of use. Free components can become proprietary if their license is not copylefting: of course this will happen only through a process by which they are significantly « extended » in functionality, so that the non-free version will appear more appealing to some users than the free version, or will simply be forced on them. In addition, since all patent offices and some case law have accepted software patents, the use of some software or standard becomes more and more governed by patent rights. In countries such as the US that have weaker systems of protection for interoperability in their copyright legislation, even terms of use of descriptions of formats and protocols can be used for achieving proprietary control.

Let's now imagine that we live in a world in which it is only natural for companies to place short term profit relatively high in their agenda, where some would like to maximise their control on enlarged markets, or where companies can be bought by others, or be auctioned after going bankrupt. There is a deadly risk that the free character of our software component will be brought to nothing in terms of practical use through a conjunction of embrace and extend strategies or other forms of proprietarisation of some well chosen components or layers on which this practical use depends (at random: OS security layers, DRM, authentication and Net identity, PKIs, compression codecs, some streaming media server functionality). Does the GPL make it impossible? Not quite, but it makes it less certain and easier to fight. It creates a cluster of inter-supporting unproprietarisable components. It forbids some changes in licenses (for the GPL-ed components[1]), and makes other changes immediately visible as agressions against the common infrastructure. If standards have built-in rules for royalty free licensing for any intellectual property that is necessary anywhere in the full chain of their usage (of course without discrimination against copylefting licenses), the existence of the common infrastructure at this level is strongly protected. This is precisely why some proprietary players want to change these rules where they exist and do not want to adopt them elsewhere.

With the broader view, GPL-copylefting might[2] make some usage of the libre software impossible (the bottom green rectangle of the right column of our diagram). But why would such a usage be impossible for GPL-ed software? This must be precisely because there exists some intellectual property necessary for using a GPL-ed version in practical applications, and the owner for instance does not want to grant a non-revocable royalty-free patent license. Or because the user intends to construct a derivative work for which a similar restriction exists. But then, if this software is released under a non-copylefting license, the same intellectual property can be used to re-proprietarise an extended version of it. What we gain from copylefting is a much greater degree of safety and sustainability of freedom, continued co-operation and usability within the full range of activities for which it was intended or later proved useful. Of course some companies will argue that by "preventing their usage" (as licensees), we deprive them from using results of publicly funded research for which all taxpayers have paid, or that we open such usage only at the expense of stripping them from legitimately acquired intellectual property. But this is not true, they are only forbidden to use these results in ways that open the prospect of abuse.

It all breaks down to one simple question: is it legitimate in the digital world for governments to support and protect the existence of commons, of public resources that can not be appropriated? If one thinks governments are only servants of economic players, and should serve them regardless of what are the effects on society, culture and the economy as a whole, one can accept the prospect of enclosure of essential public resources. If one thinks government should serve citizens and the public good, including of course creation of economic wealth by companies, then it is legitimate to contribute to a rich and protected commons domain.

The question of whether the digital commons needs to be protected against re-proprietarisation can be tackled from another angle. For physical commons, made of exhaustible resources (water, grazing land, for instance), it is clear that some protection is necessary against appropriation or misuse. But some would claim that this obligation would no longer exist for intangibles, for commons that are made of resources whose usage does not diminish or even augments the value. The very contribution of our framework is to show that as soon as components of the digital commons are inter-dependent, are developed one from the other, and that appropriation of one can lead to control on others, then one also needs a protection of these components against appropriation.

This leaves a wide open issue on identifying the scope of this necessary protected commons domain, for today and for tomorrow, but it is essential to address this issue with the right premises. Can any one doubt that the core infrastructure of information society applications, the exercise of such basic rights as accessing government services, expressing oneself and making one's opinions known to others, accessing, producing and evaluating knowledge, mastering the new tools of critical thought and information exchanges, entering into an economic activity as an independent provider, belong to the most minimal definition of necessary protected commons?

If one follows the path of this analysis, one is led to conclude that GPL copylefting serves the public good especially for any software that plays or may later play a critical role in the activities of an information society. For publicly funded efforts the consequence follows immediately: copylefting licenses are clearly to be preferred. For privately funded efforts, of course, it is obvious that the choice belongs to those who made the effort. It can be for them a difficult choice between some short term benefits of non-copylefting in terms of take-up, and the user trust effects of copylefting, as well as the long term sustainability of this approach. In practice, a company choosing non-copylefting licenses (either for formerly developed software or for software being developed) takes a strong risk in terms of free riding proprietary competitors. Dual licensing that establishes a clear relationship between two types of dissemination, and time-decaying licenses transforming themselves automatically from proprietary to GPL after a stated and short period are - all things considered - much clearer and trustable approaches to short-term reward. The main reason for discussing the case of privately funded effort in the context of this paper is that in practice, software will often result from a mixture of public and private funding. When this is the case for infrastructure components, the public funding part can only be justified if a GPL-ed dissemination path is at least left open.

[1] GPL-ed software can be re-proprietarised only by its own copyright owner, and provided that it owns the copyright on the full derivative work. In practice, this reduces the risk considerably.

[2] « Might » because non-copylefting licenses are often chosen purely for reasons of being less restrictive in their wording, even when in practice the usage range authorised by GPL-licensing would be quite similar.