



Free and Open Source Licensing

White Paper
December 2006

Chapter 1

Executive Summary

About this Series

This free and open source series of papers provides straightforward explanations of open source topics. The intention of the series is to simply explain complicated subjects like licenses, patents, and business models in easy-to-understand points. These papers present an overview of Sun's perspective on the topics. The framework for this and future papers is based on the "Zen of Open Source", a keynote delivered by Simon Phipps, Sun's Chief Open Source Officer.

Thanks to Dr Richard Stallman of the Free Software Foundation for guidance with this paper.

Keep in mind, nothing should replace the advice of a good lawyer when it comes to software licensing and patent grants.

Introduction

Within free and open source software communities today, licenses have proliferated to the point that, at the time of writing, there were nearly sixty licenses approved by the Open Source Initiative as open source licenses. From Sun's perspective, the large number of licenses can be streamlined into three main categories based on the attributes of the licenses. The existence of dozens of licenses poses an undue challenge for someone not intimately familiar with open source, making it difficult to understand the characteristics of each license and the distinctions between them, much less begin to determine how to choose a license. It can make it nearly impossible to answer essential questions, like:

- Exactly how does an organization make its decisions about free and open source licensing?
- To what extent are an organization's choices influenced by specific projects?
- How are licensing choices influenced by the ways in which communities associated with specific projects define themselves?

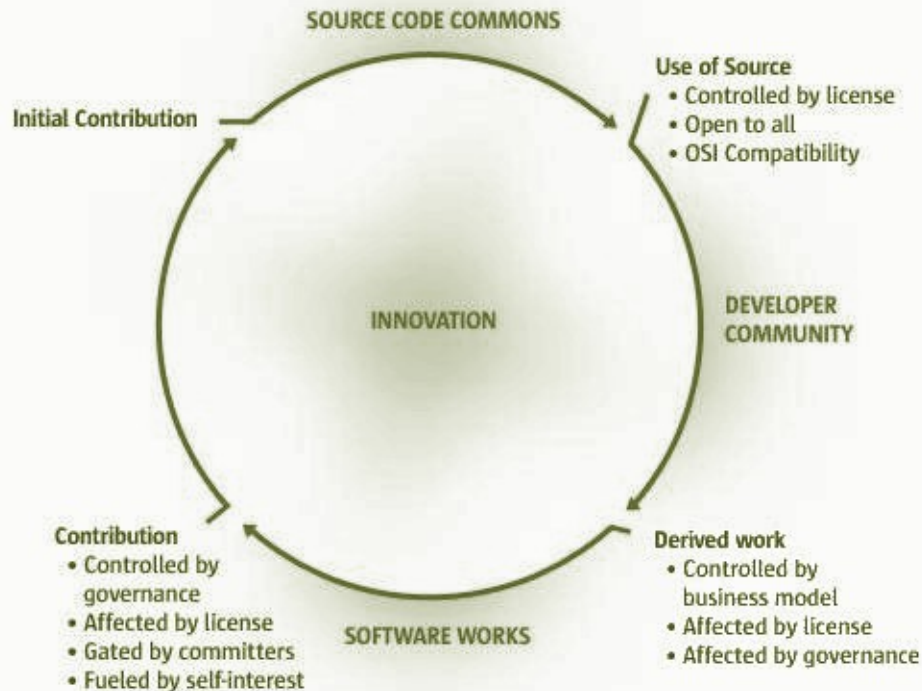
The answers to these questions are crucial to understanding the terms on which an organization participates in the open source community of communities. They are also essential to understanding the effects of an organization's licensing decisions on others.

Sun's categorization makes it possible to more easily describe the differences among license types. This in turn makes it easier to explore the underlying principles that increasingly guide Sun as it makes licensing decisions, to explain how those decisions affect the nature of Sun's participation in the free and open source communities it joins and hosts, and to suggest how Sun's decisions might inform the choices made by those who seek to participate with Sun in open source activities. Aside from this white paper, Sun supports efforts to reduce complexity in open source licensing decisions. For this reason, Sun is an active participant in the OSI License Proliferation Committee's work¹.

Please keep in mind as you review this material that choosing a free or open source software license can often be a complex process that, because of your specific needs, is best undertaken with proper legal guidance in the context of the objectives of your community. This brief explanation of Sun's philosophy and approach to free and open source licensing is by no means intended to take the place of the advice of professional legal counsel.

¹<http://www.opensource.org/docs/policy/licenseproliferations.php>

Virtuous Cycle Model of Open Source



Sun's approach to free and open source licensing is based on the concept of a *virtuous cycle*, or an endless circle in which developers share a source code commons, create derivative works from that code, and then contribute back to the commons in the form of innovation.

Within this cycle, there are variables around how developers structure their code how derivative works created from the code commons are subsequently licensed. These values that these variables take shape the community that forms around the use of code and reflect the kind of licensing that is associated with that use.

The key to making decisions about open source licensing for software development is to understand what kind of community is desired and then to move in the direction of the licensing that is most likely to promote that type of community. This is how Sun has proceeded in choosing which licenses to use for a variety of free and open source software projects, including OpenOffice.org, OpenSolaris™, and the communities around Sun's implementation of the Java™ standards, OpenJDK, Mobile & Embedded and Project GlassFish™.

The Premise of Licensing for Maximum Benefit

Sun's decisions about free and open source licensing increasingly seek to create the most attractive proposition for community participation, balancing a licensing philosophy that espouses choice for the developer to pursue their own interests with one that seeks to enrich or protect the commons. The remainder of this paper is devoted to:

1. defining the three major categorizations of free and open source licensing today
2. describing some of the types of licenses that are included in these categories
3. detailing the distinguishing attributes of these licenses and categories
4. delineating how Sun has made its licensing decisions by matching the goals of a particular free or open source project community with the appropriate licensing

Chapter 2

The Three Categories of Free and Open Source Licensing

Building on the earlier work of many, including the Free Software Foundation, the Open Source Initiative (OSI)² set out in 1998 to define the characteristics of a license likely to promote free and open source software development. Its “Open Source Definition”³ still provides definitive guidance on the characteristics of an open source license, and has been so successful that a cornucopia of licenses have been submitted for approval to the OSI Board. Today, there are many free and open source licenses available—nearly sixty. Sun has come up with three overall categories to contain these many different licenses. An understanding of these licensing categories provides a foundation for understanding Sun’s approach to making decisions about open source licensing for its products.

In this framework, licenses are categorized based on the extent to which a license requires a derivative work to use the same license as the code commons from which the work is derived. This property of a license has become known as “copyleft”.⁴ Copyleft, as its name might suggest, is a specific use of copyright. The owner of the copyright to specific code within the code commons may add specific distribution terms conferring the right to use, modify, and distribute that code or a derived work providing those terms are unchanged. This mechanism is what provides legal compulsion for a developer to put changes back to the source code commons.

Category A licenses, for example, place no requirements on a derivative work to use the same license as the code commons from which it is derived. Such licenses are *non-copyleft*. The licenses are wholly unrestricted with regard to the scope of license use in derivative works. A developer operating under a category A license may therefore use the commons to create virtually any work, in any way, and then use any kind of licensing he or she desires for the derivative work. The developer may, for example, use a free or open source license for the derivative work, or even use a commercial non-open source license for the work.

This is not to say that category A licenses are always entirely without conditions on the developer using the license. A category A license may in fact impose some conditions, such as requiring that the derivative work credit the original author of the code from the commons, or that the developer of the derivative work agree not to sue the original author under certain circumstances. For example, the Apache license described in the next chapter requires that end-user documentation of derivative works has in it an acknowledgment that the product includes software developed by the Apache Software Foundation⁵. Generally, however, conditions of category A licenses do not extend to mandating which license a developer uses for a derivative work. Category A licenses can be used to create an environment in which a technology becomes widely adopted by many different kinds of users and might thus be considered *market-creating* licenses.

In summary: category A licenses typically impose these requirements on derivative works.

- Unrestricted development of derivative works
- Wholly unrestricted scope of license use
- Any conditions of use are outside of any license mandate

Category B licenses, like category A licenses, allow developers to use the commons to create virtually any work, in any way. They assume, however, that these derivative works will be made up of source files that will ultimately become a single binary file. And they require that any such source file that contains code from the commons must be licensed with the same license as the commons. They do not, however, impose this requirement on files that do not contain code from the original commons; those files can be licensed in any way the developer wishes.

² <http://www.opensource.org>

³ <http://www.opensource.org/docs/definition.php>

⁴ <http://www.gnu.org/copyleft/>

⁵ <http://www.apache.org/>

Such licenses are *copyleft* licenses. It can be useful to think of category B licenses as *file-based* licenses, because the requirement to use the same license as the original commons is determined by the file in which the derivative work is contained. If that file contains modified or unmodified code from the commons, it must use the same license as the commons. If not, the file need not adhere to any particular license. In other words, you only have to apply category B license restrictions to a file that contains the original code or modifications to it; your own independently-developed files are not subject to any such requirement. This can occasionally raise the question of whether a file you create is itself a modification and therefore subject to the license requirements. An attorney is your best guide in making such determinations. Category B licenses foster community contribution to the commons in which the licensed files reside and might thus be considered *community-fostering* licenses.

*In summary:*The following summarizes the characteristics of category B licenses in terms of licensing requirements imposed on derivative works.

- Unrestricted development of derivative works
- File-based licensing
- Files derived from category B-licensed commons must use the same license for source files
- Files not derived from category B-license commons may use any license

Category C licenses, like category A and category B licenses, allow developers to use the commons to create virtually any work, in any way. And, like category B licenses, they require any file that contains code derived from the commons to be licensed under the same license as the commons. However, they also require that any file, regardless of code origin, which is combined under certain circumstances with the commons file must be licensed under the same license as the commons. Ideally, the effect of this is to make as much free software as possible available publicly, and in a practical way, through the commons.

While category B licenses may be termed file-based licenses, category C licenses may be termed *project-based* licenses because the effect is to license all files in the project under one license. This characteristic of category C licenses makes them a *strong copyleft*. If a file is derived from a category C-licensed commons, the entire work must be licensed under that same category C license. Category C licenses may be held to reduce the scope of competition on software features by ensuring all are able to use any source code developed in connection with commons in which the project resides.

In summary: The following summarizes the characteristics of Category C licenses in terms of licensing requirements imposed on derivative works.

- Unrestricted development of derivative works
- Project-based licensing
- Files derived from category C-licensed commons must use same category C license
- Under certain circumstances even files not derived from category C-licensed commons must use category C license if they are distributed with code under a category C license.

Each of these license categories represents a viable set of choices with which to meet different needs for open source development and licensing. The next chapter describes some of the types of licenses that fit into these categories and the types of communities for which they may be appropriate.

Chapter 3

Types of Licenses Representing the Three Major Categories

The many free and open source licenses that are available today generally fit into several types that represent the categories described in the previous chapter. Here are a few significant examples.

Category A: Non-Copyleft Licenses

Category A licenses generally represent the academic style of licensing of which the archetype is the revised Berkeley Software Distribution (BSD) license. The original BSD license was developed by the University of California to distribute its software. Today, probably the best-known license within this category is the Apache License version 2⁶. Such licenses can be used by anyone to create any derivative work for any purpose, without any requirement to contribute back to the commons from developments associated with that derivative work.

Category B: Copyleft Licenses

Category B licenses include the vast majority of the many free and open source licenses in existence today. These are licenses that have been created based on the Mozilla Public License (MPL)⁷ that emerged from the development of the original Mozilla browser in the late 1990s. The MPL was created to address a desire to ensure that improvements made their way into the software commons while still keeping open the options for individuals to explore other avenues associated with the software. Sun has most recently created a Mozilla-style license called the Common Development and Distribution License (CDDL), which it is currently using as the licensing platform for the OpenSolaris project⁸ and the Glassfish project⁹. If you would like more information about CDDL, the license is available for review online¹⁰.

Category C: Strong Copyleft Licenses

The best-known of the Category C licenses, (GNU-style) licenses, is the GNU General Public License, or GPL¹¹. The GPL has been the basis for much freely available software in the world today, which has made it the object of praise and concern, depending on your perspective. The GPL has served as a bulwark in maintaining the continued open availability of software works, although some hold that the GPL is too restrictive for their specific commercial purposes, as it may be considered to preclude certain proprietary business models.

It is also worth noting that a revision of the GPL is in progress at the time of writing¹²: version 2 of the GPL has been in use for 15 years, and version 3 seeks to address issues that have emerged subsequently, such as DRM technologies, issues around software patent litigation, and compatibility with other free software licenses.

⁶ <http://www.apache.org/licenses/LICENSE-2.0.html>

⁷ <http://www.mozilla.org/MPL/MPL-1.1-annotated.html>

⁸ <http://www.opensolaris.org/os/>

⁹ <https://glassfish.dev.java.net/>

¹⁰ <http://www.sun.com/cddl/>

¹¹ <http://www.fsf.org/licensing/licenses/gpl.html>

¹² <http://gplv3.fsf.org/>

Please remember that the information presented here is not intended to take the place of the knowledge or advice of professional legal counsel.

Summary table:

<i>Category A</i>	<i>Category B</i>	<i>Category C</i>
Unrestricted, non-copyleft	File-based copyleft	Project-based copyleft
Create any work	Files derived from commons must use same license	All files in project must use the same license as the commons if any one file from the commons is used in the project
No restriction on licensing	Files added may use any license	Code added to the project must also use the same license as the commons
Marketplace-creating	Community-fostering	Commons-protecting

Chapter 4

Attributes and Effects of License Types on Free and Open Source Software Communities

To return to a premise stated earlier in this paper, Sun increasingly makes decisions about free and open source licensing for its products based on its view of open source as a virtuous cycle. The extent to which it is incumbent upon a developer to contribute any innovations based upon a source code commons back to that commons is influenced by the license. All contributions may be at the developers discretion, or it may be the case that any innovation based upon the commons must be contributed back before it may be further distributed. Sun considers that it is typically rational to contribute innovation back to a source code commons, however, these differences may lead to wholly different community dynamics, which Sun also takes into account in its license selection.

Complete Developer Discretion, Minimum Community Protection: Category A

Category A licenses, or non-copyleft licenses, offer developers unfettered choice in determining how to structure their code and license their derivative works, with no requirements imposed from the commons as to which category or type of license will be used. It still may be the case that contributing back to the commons may be a rational, or self-interested, activity. A developer who has a long-term interest in the health of the source code commons and the community that manages it, not to mention in the long-term relevance of their own code, may find it advantageous to put their code back into the commons. However, as there is no compulsion to contribute, the possibility exists that those deriving forks from the code commons will choose not to enrich that commons with bug fixes and other improvements. Anyone is permitted to use software licensed under the revised BSD license, for example—even to create proprietary works—but no one is required to add anything back to the BSD commons in return.

Restricted Developer Discretion, Maximum Community Growth: Category C

At the other end of the spectrum from category A, category C licenses restrict the developer's ability to choose a license for themselves for their derivative works, as the license may dictate precisely which license must be used for such works. This leaves developers with a restricted capacity to determine their own business models and licensing for the works they create. From one perspective, this is a trade-off: interests in proprietary software business models may find themselves unable to operate within such a model, but all those who do choose to participate in the community will typically contribute to the commons. This promotes the growth of the commons for the community to a greater extent than any other category of license. For many category C licenses, such as the GPL, this outcome is not an expedient, but merely the concomitant effect of protecting the freedom of the end user.¹³

Balancing of Developer Inclusion and Commons Protection: Category B

Category B licenses have the capacity to hold in balance the ability to attract a variety of interest to a community with the strength and longevity of a source code commons. They allow developers to create software inventions based upon the commons and license them as they see fit, but at the same time they compel developers to improve the commons from which their works are derived: any changes to the commons itself must be returned. Again, it may well be rational to contribute additional inventions back to the commons, regardless of obligation, but this will be at the developer's discretion. This balanced approach is reflected in many new open source licenses.

¹³ <http://www.gnu.org/philosophy/free-sw.html>

Chapter 5

Sun's Approach Exemplified

Sun's philosophy of free and open source licensing is not to necessarily favor any one license or category of license but to make decisions that match the project or community with the license that best supports it. Here are a few examples of Sun's decision-making process with regard to open source licensing for various Sun projects over the years.

The Jini™ Community: Licensing for Unlimited Developer Freedom and Business Models

The Jini community¹⁴ is a pioneering software community that Sun created in the mid-1990s. After using a non-open source license for many years, the community recently decided to choose a free /open source license. As a loose-knit community deriving many different works from the same code commons, the Jini community placed a high value on the freedom of developers to innovate around its code base and choose business models for derivative works virtually without restriction. Accordingly, the community chose the Apache Software License version 2. Such a Category A license is entirely in keeping with the value placed on developer choice within the community rather than on the growth or protection of the commons.

OpenSolaris: Licensing to Balance Enterprising Development and a Strong Commons

The OpenSolaris community that is developing upon the a free and open source Solaris source code base wanted to have a strong community focused around the Solaris code commons — but also wanted to afford developers the ability to mix in code of other licensing origins in order to create the most innovative derivative works possible, without restrictions on the licensing of those works. This desire to balance a commercial incentive to innovate with the desire to enrich the commons led to the choice of the Category B CDDL, which Sun developed from the MPL, expressly to encourage the creation of derivative works for commercial purposes while still strengthening and protecting the commons.

Project Looking Glass: Licensing to Protect Non-Commercial Projects

At the other end of the spectrum from the Jini Community is Project Looking Glass¹⁵, a project that the Sun launched to conduct research into three-dimensional user interfaces. In this case, the community was a largely non-commercial one devoted primarily to experimenting with and making decisions about the future of the Looking Glass code base. In keeping with this focus on non-commercial values, Project Looking Glass elected to license its code under the GPL. This widely used Category C license makes sense for a community in which the evolution of the commons is of far greater value than the development of derivative works from the commons.

OpenOffice.org: Licensing for Joint Development and Code Improvement

OpenOffice.org is the open source project through which Sun Microsystems has released the technology for the popular StarOffice™ Productivity Suite. The OpenOffice.org project is licensed under the GNU Lesser General Public License (LGPL)¹⁶, a special version of the Category C GPL that addresses the linking of proprietary applications with an original work. The OpenOffice.org project allows proprietary macros, plug-ins, and documents to be linked with the project under the LGPL in a Category B manner.

¹⁴ <http://www.jini.org/>

¹⁵ <https://lg3d.dev.java.net/>

¹⁶ <http://www.fsf.org/licenses/lgpl.html>

Chapter 6

Conclusion

Sun takes a license-neutral, context-based approach to free and open source licensing. It is probably fair to say that looking forward, the majority of Sun projects will tend not to use category A licenses. This is because Sun's involvement in free and open source communities favors strength within the community and a culture of contribution to a source code commons.

Sun also recognizes that the proliferation of open source licenses can represent a problem for the open source developer. That is why Sun supports the work of the OSI's license proliferation committee and, whenever possible, will use existing, OSI-approved licenses.

For these reasons Sun will tend to favor category B and C licenses, and specifically the CDDL and the GPL. To learn more about Sun's involvement in and approach to free and open source, visit www.sun.com/opensource.

© 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, CA 95054 USA

Sun, Sun Microsystems, the Sun logo, GlassFish, Java, Jini, OpenSolaris, Solaris and StarOffice are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are trademarks of their respective owners.

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.