

Open Source Licensing Models

The Implications of Using Open Source in the Enterprise

William Shampton, November 2009

Over the last few years, free and open source software (FOSS) has become more and more widespread in the modern enterprise. Cost benefits, continued improvements in software and associated services, and decreased dependence on vendors are some of the driving factors for this trend. Often, however, business and IT decision makers do not take open source software into consideration. Concerns about legal issues and uneasiness about the involved licenses are mentioned frequently as perceived risks, but an analysis of common FOSS licenses reveals that risks for an enterprise making use of FOSS are minimal or actually comparable to using commercial software.

Open source is defined by the Open Source Initiative as: 1) software that must be redistributed to anyone else without any restriction; 2) source code that must be made available (so that the receiving party will be able to improve or modify it); 3) a license that can require improved versions of the software to carry a different name or version number from the original software.

Today there are more than 250,000 open source projects. Well-known projects include the Linux operating system, the Apache Web server, the MySQL database, and the JBoss application server.

Free and open source software comprises components and solutions whose source code is available and can be used, changed, and distributed to other users following commonly agreed-upon rules. These rules are defined in the license the software is distributed under, like ordinary proprietary software. A FOSS license is, of course, different: rather than retain or expand the software developer's rights, it gives them away. It is generally intended to protect the philosophical freedoms granted to software users, and re-enforce the perceived strengths of a peer-to-peer development model.

Problem:

Business and IT decision makers may perceive risks associated with free and open source software. This is often due to legal concerns, and uneasiness surrounding open source licenses.

Solution:

An understanding of the prevalent license models shows that the use, change, and distribution of open source software has become easier over time. In this paper, we showcase sound principles that highlights the safe integration of open source licensing models into enterprise.

Benefit:

Organizations are able to take advantage of the cost benefits, improvements, and vendor independence made possible by leveraging free and open source products and solutions without undue concern for legal implications.

Practical Implications

According to leading industry analysts, the majority of enterprises use open source as part of its mission critical applications. For the most part, enterprises using FOSS should have few concerns about licensing for the following key reasons:

- FOSS licenses allow the software to be executed without restriction.
- Under FOSS licenses, source code can be modified without restriction if the resulting software is being used internally.
- Some FOSS licenses place requirements for re-licensing and publication on the enterprise only if it plans to distribute the software to its customers. The remaining FOSS licenses have no such requirement.
- If buying packaged free or open source software, or a system that contains FOSS (for example, Red Hat Advanced Server or HP/UX), then Red Hat or HP EULA is responsible for the primary license and all other third party license concerns are left to the vendor.

As you can see, there is no single open source license. To date, more than 35 exist. There are, however, three primary license families:

- **Academic licenses (MIT Athena, Berkeley, and Apache)**
- **Free software licenses (General Public License and the LGPL)**
- **Mozilla-style licenses (Mozilla and the IBM licenses)**

They may be broadly categorized into the following types:

- (1) Those that apply no restrictions on the distribution of derivative works (non-protective licenses do not protect the code from being used in non-open source applications).
- (2) Those that do apply such restrictions (known as protective licenses, because they ensure that the code will always remain open and free).

Shared source licenses, such as the Microsoft Reference License (MS-RL), have some similarities with open source but are not compatible with the open source definition and are not covered here.

Academic Licenses (MIT Athena, Berkeley, and Apache)

In the 1980's, the University of California, Berkeley, was working on early UNIX systems and acted as a repository for the collaborative developer community. The university created a license for its work, the Berkeley license, to encourage new research and use of the software created. The Berkeley license was also the basic model for the MIT Project Athena license (used for the X11 windowing technology), and also the model used by the Apache community at the University of Illinois at Urbana-Champaign. In 2004, the Apache 2.0 license was released. It is a complete rewrite that accounts for contributions and patents (and is more complex), but it remains true to the foundations of the Berkeley-style license.

The Berkeley-style license:

- Allows users to do anything with the software, including extending and selling.

- Does not require any derived software to be under the same license or that the changes be published, which allows the inclusion of software with this license in proprietary products.
- Requires that attribution be given for the work and copyrights maintained.
- Denies any warranties (express or otherwise), similar to a proprietary EULA.
- Is a non-protective license.

Example: A developer might write a web-based application leveraging the Lucene search engine. As Lucene is released under the Apache 2 license, the developer she would be required to retain attribution for search functionality to Lucene (she cannot not claim as her own work). No other restriction would be placed on sales or distribution of her application and she could license their work as she saw fit.

Free Software Licenses (General Public License and the LGPL)

The non-profit Free Software Foundation (FSF) was formed in 1985 to promote the ideas that a program's source code should always be available, and a user can always fix and extend the software without restriction. The General Public License (GPL) was created by the FSF to further this belief. Many of the most important FOSS programs of the past twenty years are licensed under the GPL, including the Linux operating system, the GNU Compiler Collection (GCC), the MySQL database, and the JBOSS application server. Many vendors (including software vendors) use and develop software licensed under the GPL.

The GPL license:

- Does not allow distributed derivatives of GPL-licensed software to be re-licensed under other (possibly proprietary) terms.
- If one uses any GPL-licensed source code in a program and distributes the program, the entire result (including the source code) becomes subject to the GPL.
- Denies any warranties (express or otherwise), just as a proprietary EULA does.
- Is a protective license.

The Lesser GPL (LGPL) was developed later to account for software libraries. The LGPL allows a library to be licensed, but not the entire application (enabling it to remain closed), while, if distributed, changes to the library itself must be published under the LGPL.

An important point to note is that the requirement to maintain GPL licensing on derivatives is only in effect on distribution of the software, not on using it.

Example: A developer creates a desktop application that uses the MySQL database for local persistence. If he chooses to distribute his application, it must be under the GPL license and his source code would have to be released under the same license. If he wishes to use it himself, no such restriction applies.

Mozilla-Style Licenses (Mozilla and the IBM Licenses)

In the late 1990s, Netscape published the source code to their browser and began to build a community of developers around it: the Mozilla project. The license they created for this project

was the Mozilla Public License (MPL), one of the first licenses created by a corporation. We've seen a rapid increase of open source software licenses based on the Mozilla license as companies wishing to create software communities as a business tool invariably want to change the MPL language to be less Mozilla-centric.

The MPL:

- Requires derivatives of an MPL work to be licensed under the MPL, which is similar to the GPL.
- Enables MPL-licensed works to be combined with other software and re-licensed into a larger work, which enables the development of proprietary software similar to the academic licenses.
- Mentions patent rights relevant to the licensed work.
- Denies any warranties (express or otherwise), just as a proprietary EULA does.
- Is a protective license.

Example: A developer might create an application that allows users to view movies and photographs on a dedicated set-top box. Portions of the application rely on the Firefox web browser, but the browser's source code has been modified to fit the need of the application. In order to distribute her application, the developer must release the changes she made to Firefox under the MPL. Her proprietary work, and the work as a whole, may be licensed as she sees fit.

Conclusion

As we see from the data above, the practical implications of using an open source program on a day-to-day basis ordinarily poses few, if any, legal implications for the user.

Anyone who uses open source software has “the freedom to modify the program or to incorporate it into other programs.” As long as your organization uses only the derived work internally, the FOSS license should not affect you at all.

You may also use FOSS to provide services to the public without applying any measures involving permission to reproduce, adapt, or distribute the work. For example, an organization can modify open source content management software and place it on a server to operate a commercial Web site. The organization is not publishing or distributing the modified software to the public, so the modifications do not fall under reproduction, adaptation or distribution provisions.

According to Gartner more than 80 percent of companies already use open source today. Most of them have 50, 100, or more open source components and systems in production. This is partly an ingredient of commercial products, employed as a component, or integrated as infrastructure. Decision makers at these companies have arrived at an understanding of the prevalent license models and know that the use, change, and distribution of open source software has become easier over time and follows sound principles that allow safe integration into the enterprise.

Sources:

- The Open Source Definition (<http://www.opensource.org/docs/definition.php>)
- The Free Software Foundation definition of free software (<http://www.fsf.org/licensing/essays/free-sw.html>)
- Open Source Initiative approved licenses: (<http://www.opensource.org/licenses/>)
- Understanding Open Source Software - by Red Hat's Mark Webbink, Esq.: (<http://www.groklaw.net/article.php?story=20031231092027900>)

About the Author



William Shampton, Technical Architect

William has over 10 years of experience in web application development, utilizing a wide array of well-recognized open source technologies. He is always striving to enable breakthrough improvements in the competitive, sustainable delivery of high value, high quality software. William joined Razorfish in 2007 to advance his quest. He attended The University of Texas at Austin, fell in love with the town and has been there ever since. He is not an attorney.

Email William your comments or suggestions: William.Shampton@razorfish.com

About Razorfish™

Razorfish creates experiences that build businesses. As one of the largest interactive marketing and technology companies in the world, Razorfish helps its clients build better brands by delivering business results through customer experiences. Razorfish combines the best thought leadership of the consulting world with the leading capabilities of the marketing services industry to support our clients' business needs, such as launching new products, repositioning a brand or participating in the social world. With a demonstrated commitment to innovation, Razorfish continues to cultivate our expertise in Social Influence Marketing, emerging media, creative design, analytics, technology and user experience. Razorfish has offices in markets across the United States, and in Australia, China, France, Germany, Japan, Spain and the United Kingdom. Clients include Carnival Cruise Lines, MillerCoors, Levi Strauss & Co., McDonald's and Starwood Hotels. Razorfish is part of the Publicis Groupe (Euronext Paris: FR0000130577) VivaKi organization. Visit razorfish.com for more information. Follow Razorfish on Twitter at [@razorfish](https://twitter.com/razorfish).

Razorfish

821 2nd Avenue, Suite 1800
Seattle, WA 98104
Phone: 206.816.8800
Fax: 206.816.8808