Open Source Evolution v. 2019

By: Paul H. Arne[1,2,3]

Open source software has come a long way. Started in the early 1980s, the open source movement as a means of software development and distribution has exploded.[4] Open source software is widely used; it is virtually impossible for corporate america to not use open source software.[5]

This article discusses some of the new developments in the ongoing evolution of open source licenses and the business models that use them. Specifically, these new developments are the Server Side Public License, the Commons Clause, and the GPL Cooperation Commitment. To put these developments in context, however, some background is needed.

1. Background

A. Open Source and Copyleft

Originally, the open source software movement was focused much more on software developers being free to use and modify code than it was on businesses making money. This is especially true for those open source licenses that have "copyleft" provisions ("copyleft licenses"). Generally speaking, copyleft provisions are:

1. the source code of the software **or any derivative work of the software** must either be distributed at the same time or a mechanism for obtaining the source code for no more than a nominal fee must be established;

2. the user must be allowed to modify the software and any derivative without additional charge; and

3. the software and any derivative must be distributable by the user without additional charge.

---

[2] Paul is the senior partner and chair of the Technology Transactions Group of Morris, Manning & Martin, L.L.P., as well as the founder and chair of the firm's Open Source Practice Group.

[3] This article does not create an attorney/client relationship with you and does not provide specific legal advice to you or your company. Certain legal concepts have not been fully developed and certain legal issues have been stated as fact for which arguments can be made to the contrary, due to space constraints. It is provided for educational purposes only.

[4] Black Duck Software, Inc., a popular source of information and services related to open source software has over 2 million open source projects in its database. See https://www.blackducksoftware.com/top-open-source-licenses (last reviewed on November 26, 2018).

[5] Black Duck Software, Inc. reports on a 2015 Gartner survey, where 99 percent of all responding organizations reported that they used open source software. See https://www.blackducksoftware.com/top-open-source-licenses (last reviewed on November 26, 2018).

(i) What's a Derivative?

For users of open source software governed by copyleft licenses, the copyright law question of whether one piece of software is a derivative work of another becomes critically important. This issue is especially difficult when software programs are created using separate executable programs—in separate electronic files—that talk to each other, which is the way software is typically developed.

Logically, for software to work with other software, each must communicate with the other. For one piece of software to invoke and use another, the means of communication—such as the syntax of the communication, the names of the commands invoked, etc.—must be in a certain form. This common language of communication, typically called "interface information" or an "API", must be contained in both pieces of software. In programming parlance, having one piece of software "call" another piece of software to perform certain functionality is called "dynamically linking" the two pieces of software.

Frequently, the interface information is sufficiently robust as to be "expressive" in the context of the idea-expression dichotomy under copyright law. If the interface information is expressive, as opposed to an "idea," then it is copyrighted. The placement of such interface information from one piece of software into a second piece of software would therefore make the second piece of software a derivative work of the first software package.[6,7]

Given the above information, it is not particularly surprising that the Free Software Foundation, which manages the various versions of the GNU General Public License[8] (generically, the "GPL") states the following in its FAQ.

> Linking a GPL covered work statically or dynamically with other modules is making a combined work based on the GPL covered work. Thus, the terms and conditions of the GNU General Public License cover the whole combination.[9]

While one can debate whether this assertion is always true, you can see the logic that the use of another piece of software for functionality, with its necessary use of interface information, creates at least the risk that a derivative work has been created. Because the creation of a derivative work is one of the exclusive rights granted to the copyright holder, the copyright holder has the right to place conditions on the creation of derivatives, which is where copyleft licenses come into the picture.

---

[6] An example of how this applies can be found in Oracle Am., Inc. v. Google Inc., 750 F.3d 1339 (Fed. Cir. 2014), where Google, in creating its Android operating system, copied the names and syntax of the Java methods, classes and packages in Java, without copying any functional code (other than a small amount which was addressed separately), and was found to violate Oracle's copyright in the Java development environment.

[7] The explanation in this section should not be considered a full analysis of the legal issues here. Only enough information to create a context for the new developments discussed in this article is provided. The actual analysis is much more robust.

[8] Discussed in Section 2 below.

[9] Frequently Asked Questions about the GNU Licenses, https://www.gnu.org/licenses/gpl-faq.html. Does the GPL have different requirements for statically vs dynamically linked modules with a covered work?

B. Evolution of the General Public License

The oldest, most well-known and most-used open source software license containing copyleft provisions is the GNU General Public License. The GNU General Public License, Version 2 ("GPLv2") was first published in 1991 by the Free Software Foundation[10] ("FSF"). In the early 1990s, the fastest modem dial-up access to the internet was 28.8 kilobits per second. By comparison, as of 2016, residential cable connections to the internet max out at about 2 gigabits per second,[11] a 69-fold increase in speeds. Not surprisingly, the implications of cloud computing and software-as-a-service offerings were not addressed in the language of GPLv2. In GPLv2, the mechanisms used to insure that software could be always used and modified by programmers—the copyleft provisions—are triggered by distribution. If there is no distribution, there are no copyleft requirements placed on the licensee.

The GPL was modified in 2007, resulting in the creation of the GNU General Public License, Version 3 ("GPLv3").[12] By that time, cloud computing was clearly contemplated, and the ability to exploit open source software without distributing it was viewed by many in the open source community as a "hole" in the GPL that needed to be plugged. Other users of open source, who were using open source software in reliance that the copyleft requirements are triggered only on distribution, were opposed to any change to the GPL that would require cloud providers to stop using GPL-licensed open source. These users were not willing to adopt a new version of the GPL that required copyleft obligations absent distribution.

Accordingly, the Free Software Foundation not only produced a revised version of the GPL, GPLv3, it also produced a new license: the GNU Affero General Public License, Version 3[13] ("AGPL"). GPLv3 retained distribution as the trigger for when the copyleft requirements would apply. The AGPL changed this trigger. The AGPL is very similar to the GPLv3, except that it added a new clause (section 13):

> Notwithstanding any other provision of this License, if you modify the Program, your modified version **must prominently offer all users interacting with it remotely through a computer network** (if your version supports such interaction) an opportunity to receive the Corresponding Source of your version by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software.[14] (Emphasis supplied.)

[10] See https://www.gnu.org/licenses/old-licenses/gpl-2.0.html (last reviewed on November 26, 2018).

[11] See https://www.ncta.com/whats-new/a-history-of-speed-as-the-internet-turns-25 (last reviewed on November 26, 2018).

[12] See https://www.gnu.org/licenses/gpl.html (last reviewed on November 26, 2018).

[13] There is no version 1 or 2.

[14] GNU Affero General Public License, at Section 13.

C. Evolution of Open Source Business Models

Many companies have figured out a way to be successful even when the source code of their flagship product is provided without a license fee. The first companies with open source business models typically provided services that were needed but not available due to the lack of a single vendor for open source software. They provided training, support, error correction, hosting and other services for the open source software. A good example of this kind of business model is Red Hat, Inc., which began its operations as a provider of services around Linux, a very popular and very robust operating system licensed under GPLv2. Red Hat was recently sold to IBM for approximately $34 billion.

Other companies have used a "dual license" strategy, offering the software as a free, GPL-licensed download for certain versions of the software, but also offering either a more robust or better-supported version of the software for a fee using a proprietary license.[15] An example of this kind of licensing strategy is Oracle's licensing of MySQL, which offers a free, "community" edition of the software and multiple proprietary license versions that include various other pieces of software and services.[16] Other companies provide open source software using a copyleft license to the public in hopes that it will promote a standard that is advantageous to other, non-open source products of the company.

D. Sufficiency of AGPL

From a programmer's standpoint, the AGPL would seem to solve all problems. If any software is distributed, or the functionality of the software is provided over the internet, the developer licensee has the right to receive source and do anything with it he or she desires. The AGPL is also useful for those companies that fear a competitor will take their open source products, improve on them, and then compete against them with SaaS offerings. The AGPL can be a useful license for those who worry about competition from an "improved" product that is provided on a SaaS basis.

It is not surprising, however, that the AGPL did not contemplate a different form of business competition: a SaaS software provider that offers the functionality of software as a service, without modifying the software itself. At the time of creation of the AGPL, few people would have predicted the dominance of Amazon Web Services, Microsoft Azure, the Google Cloud, and other cloud infrastructure providers. Because of their dominance, some software providers believe that cloud providers can obtain much of the economic benefit of providing open source software without contributing much, if anything, to the cost of developing or improving the open source software itself.

2. MongoDB's Solution: the SSPL

A. MongoDB generally

MongoDB, Inc. is the provider of a next-generation database engine. According to its 10K filing, MongoDB earned approximately $154 million in revenues for its fiscal year ending January, 2018. Until recently, it was licensed under the AGPL.

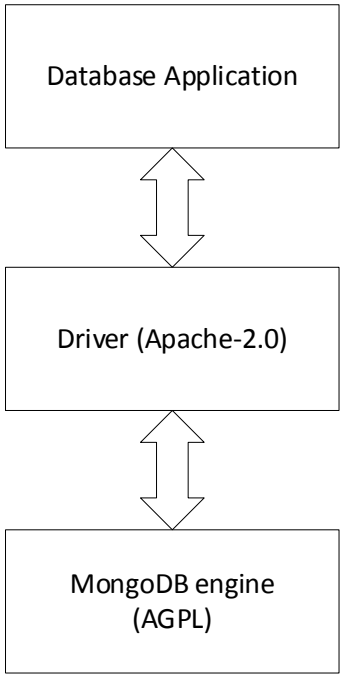
[15] There are a significant number of implications with these business models that should be addressed before undertaking this kind of business model, which are outside the scope of this article.

[16] See https://www.mysql.com/products/ (last reviewed on November 27, 2018).

If you create an application that uses MongoDB, are you also required to comply with the AGPL's copyleft obligations?[17] This is an important question, not only to a prospective user of MongoDB, but to the makers of MongoDB itself. If the use of MongoDB requires that applications built using MongoDB must also be distributed under a form of copyleft license, then fewer developers will be willing to use MongoDB as the basis for their database applications.

At first blush, it would seem that the AGPL would require applications written using MongoDB to also be licensed under the AGPL, for the reasoning described in Section 1.A(i) above. A developer of a database application would seem to need to use interface information of MongoDB in order to create the database application, resulting in the possible creation of a derivative work.

However, MongoDB has attempted to address this issue. While it is critically important for lawyers to confirm this when addressing the copyleft implications of using MongoDB, database applications apparently do not interface directly with the MongoDB database engine. Instead, they interact with database "drivers." The drivers created by MongoDB are not licensed under the AGPL, but are instead licensed under the Apache License v. 2.0[18] ("Apache-2.0"). Apache-2.0 does not contain copyleft provisions. Therefore, the design of a database application that uses MongoDB would ordinarily look like this.



The interface information used by the database application would come from the Apache-2.0-licensed driver and not the MongoDB itself, thereby significantly reducing the likelihood that the database application itself would contain interface information from MongoDB, which in turn reduces the risk that the database application will be treated as a derivative work that is subject to the copyleft requirements of the AGPL.

[17] This analysis is applicable both to MongoDB as licensed under the AGPL and the SSPL, discussed below.

[18] See https://www.mongodb.com/community/licensing (last reviewed on November 28, 2018).

B. The Server Side Public License

However, MongoDB has apparently become increasingly concerned about competition from cloud infrastructure providers, so much so that it has decided to change its open source license from the AGPL to a new license: the Server Side Public License ("SSPL"). The SSPL is mostly identical to the AGPL,[19] except that it has replaced Section 13 of the AGPL (quoted above) with the following:

> If you make the functionality of the Program or a modified version available to third parties as a service, **you must make the Service Source Code available via network download to everyone at no charge**, under the terms of this License. **Making the functionality of the Program or modified version available to third parties as a service includes, without limitation, enabling third parties to interact with the functionality of the Program or modified version remotely through a computer network,** offering a service the value of which entirely or primarily derives from the value of the Program or modified version, **or offering a service that accomplishes for users the primary purpose of the Program or modified version.**
>
> "Service Source Code" means the Corresponding Source for the Program or the modified version, and the Corresponding Source for all programs that you use to make the Program or modified version available as a service, including, without limitation, management software, user interfaces, application program interfaces, automation software, monitoring software, backup software, storage software and hosting software, all such that a user could run an instance of the service using the Service Source Code you make available.[20] (Emphasis supplied.)

As can be seen, the SSPL requires that if MongoDB is provided as a service remotely over the internet, then not only must MongoDB be provided for free in source code form, but all other software used to provide the service remotely must also be provided in source code form without additional cost, including all "management software, user interfaces, application program interfaces, automation software, monitoring software, backup software, storage software and hosting software." This is likely to be impossible for a cloud provider to do, especially given the possible license limitations of those other software programs. Even if distribution of these other software programs was legally permissible, it is unlikely that a cloud provider would be willing to do so.

[19] See https://webassets.mongodb.com/_com_assets/legal/SSPL-compared-to-AGPL.pdf?_ga=2.266285263.1712238203.1543361321-1889879843.1541635316 (last reviewed on November 27, 2=018)

[20] See https://www.mongodb.com/licensing/server-side-public-license (last reviewed on November 27, 2018).

The SSPL is applicable to all versions of the MongoDB Community Server software (i.e., the free, open source version) released after October 16, 2018, as well as any future patches of older versions.[21]

C. What Does This Mean for Applications Using MongoDB?

The SSPL states that the source code to these additional software products, called the "Service Source Code" must be made available if you "make the functionality of the Program or a modified version available to third parties as a service." The SSPL gives as an example of making the functionality of the Program available is "enabling third parties to interact with the functionality of the Program or modified version remotely through a computer network." Does this include making the functionality of MongoDB available through a database application? From the language of the SSPL, this seems possible.

However, MongoDB does not seem to intend this from the language. In its FAQ, MongoDB has the following question and answer.

> Q: What are the implications of the SSPL on applications built using MongoDB and made available as a service (SaaS)?
>
> A: The copyleft condition of Section 13 of the SSPL applies only when you are offering the functionality of MongoDB, or modified versions of MongoDB, to third parties as a service. There is no copyleft condition for other SaaS applications that use MongoDB as a database.[22]

Companies and their lawyers should therefore consider whether they are willing to rely on this statement for the interpretation of this license language.

D. Is the SSPL an Open Source License?

MongoDB seems to think so. The company is submitting the SSPL to the Open Source Initiative ("OSI"),[23] which is an organization that certifies licenses as complying with its "Open Source Definition." As of this writing, OSI has not made a determination of whether the SSPL will be certified as an open source license or not.

3. Redis Labs Solution: the Commons Clause

Redis is another popular database engine, developed by Redis Labs. The open source, "community" version of Redis is licensed under the BSD license, which is not a copyleft license. Certain modules used with Redis are licensed under the Apache-2.0 license, with the addition of a new clause: the Commons Clause License Condition v1.0 ("Commons Clause"). The concept of the Commons Clause is that it can be added to various open source licenses.

---

[21] https://www.mongodb.com/licensing/server-side-public-license/faq (last reviewed on November 28, 2018).

[22] https://www.mongodb.com/licensing/server-side-public-license/faq (last reviewed on November 28, 2018).

[23] See the answer to the question "Can you really call yourself an open source company, or describe your products as open source if you are not using an OSI-approved open source license?" in the Server Side Public License FAQ, found at https://www.mongodb.com/licensing/server-side-public-license/faq.

Redis explains the reasoning for the use of the Commons Clause as follows:

> Modern open source infrastructure software has created more value over the past decade than we could have ever imagined. Databases, orchestrators, distributed systems and other software technologies now power nearly every business on the planet; all thanks to the shared, collaborative philosophy of the open source community.
>
> However, today's cloud providers have repeatedly violated this ethos by taking advantage of successful open source projects and repackaging them into competitive, proprietary service offerings. Cloud providers contribute very little (if anything) to those open source projects. Instead, they use their monopolistic nature to derive hundreds of millions dollars in revenues from them. …
>
> Today, most cloud providers offer Redis as a managed service over their infrastructure and enjoy huge income from software that was not developed by them. … [W]e decided to prevent cloud providers from creating managed services from certain add-ons on top of Redis (e.g. RediSearch, Redis Graph, ReJSON, Redis-ML, Rebloom). These are licensed Apache 2.0 modified with Commons Clause.[24]

The Commons Clause reads as follows.

> The Software is provided to you by the Licensor under the License, as defined below, subject to the following condition.
>
> Without limiting other conditions in the License, the grant of rights under the License will not include, and the License does not grant to you, the right to Sell the Software.
>
> For purposes of the foregoing, "Sell" means practicing any or all of the rights granted to you under the License to provide to third parties, for a fee or other consideration (including without limitation fees for hosting or consulting/ support services related to the Software), a product or service whose value derives, entirely or substantially, from the functionality of the Software. Any license notice or attribution required by the License must also include this Commons Clause License Condition notice.[25]

Unlike the SSPL, which requires certain additional software to be made available in source code form, the Commons Clause simply prevents the use of the software "to provide to third parties, for a fee or other consideration ... a product or service whose value derives, entirely or substantially, from the

[24] https://redislabs.com/community/licenses/ (last reviewed on November 28, 2018).

[25] https://commonsclause.com/ (last reviewed on November 28, 2018).

functionality of the Software." Preventing the use of the software for certain kinds of uses while allowing other uses, at least with reference to OSI's definition of open source licenses, precludes any license that contain the Commons License from being treated as an OSI-certified open source license. In the FAQ for the Commons Clause, this is expressly acknowledged.

> Q: Is this "Open Source"?
>
> A: No.
>
> "Open source", has a specific definition that was written years ago and is stewarded by the Open Source Initiative, which approves Open Source licenses. Applying the Commons Clause to an open source project will mean the source code is available, and meets many of the elements of the Open Source Definition, such as free access to source code, freedom to modify, and freedom to re-distribute, but not all of them. So to avoid confusion, it is best not to call Commons Clause software "open source."[26]

The FAQ for the Commons Clause is a good read for attorneys who practice in this area. It describes how the Commons Clause is intended to allow for the offering of functionality "on top of" software licensed using the Commons Clause and to allow for the provision of consulting services. It also contains well-written distinctions among proprietary licenses, open source licenses, and the Commons Clause, as well as a criticism of the AGPL.

The Commons Clause is a second alternative way of addressing the perceptions by many that large cloud infrastructure providers are taking too much of the economic benefit of open source software, to the detriment of those who develop it.

4. GPL Cooperation Commitment

One of the differences between GPLv.2 and GPLv.3 is how they treat violations of the license, especially related to termination.

Section 4 of GPLv.2 states:

> You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.[27]

As you can see, it is absolute. Violate the license, lose the license.

[26] See https://commonsclause.com/ (last reviewed on November 28, 2018).

[27] https://www.gnu.org/licenses/old-licenses/gpl-2.0.html (last reviewed on November 28, 2018).

Section 8 of GPLv.3 is less absolute:

> You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).
>
> However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.
>
> Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.[28]

As you can see, GPLv.3 introduced the concept of a cure period into the GPL.

As open source becomes more and more mainstream, many in the open source community believe that providing a cure period is a better way to handle violations. The termination of the license automatically upon any breach may actually be a barrier to the widespread use of the software.

Red Hat has introduced a solution to this issue: the GPL Cooperation Commitment.[29] The GPL Cooperation Commitment states:

> Solely for any software for which I personally own copyright that is licensed under a Covered License, before filing or continuing to prosecute any legal proceeding or claim (other than a Defensive Action) arising from termination of a Covered License, I commit to extend to the person or entity ("you") accused of violating the Covered License the following provisions regarding cure and reinstatement, taken from GPL version 3. As used here, the term 'this License' refers to the specific Covered License being enforced.
>
> "However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

[28] https://www.gnu.org/licenses/gpl.html (last reviewed on November 28, 2018).

[29] See https://www.redhat.com/en/blog/gpl-cooperation-commitment-and-red-hat-projects (last reviewed on November 28, 2018).

> Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice."
>
> I intend this Commitment to be irrevocable, and binding and enforceable against me and assignees of or successors to my copyrights.

The GPL Cooperation Commitment is a way for companies and individuals to bind themselves to the termination provisions of GPLv.3 while still using the GPLv.2 (or Lesser General Public License, version 2.x).[30]

As of this writing, 41 companies have signed on to this commitment, including Adobe, Amazon, AT&T, Cisco, Facebook, Google, IBM, Intel, Microsoft, SAP, SAS and VMware.[31]

5. Conclusion

As technologies and business models change in the open source software space, it should come as no surprise that open source software licenses will also need to evolve.

The GPL Cooperation Commitment is a clear attempt to make software licensed under GPLv2 more palatable to businesses that need to be able to depend on that software for important business functions.

It is not clear at this point whether either the SSPL or the Commons Clause will be widely adopted to address the perception that large cloud providers are receiving more than their fair share of profits from open source software, or whether the open source industry will even collectively view this as a problem that needs a solution. As of this writing, the older of these two developments is only six months old.

Only time will tell.

[30] The FAQ for the GPL Cooperation Commitment is useful to understand the parameters of this commitment. See https://gplcc.github.io/gplcc/ (last reviewed on November 28, 2018).

[31] See https://gplcc.github.io/gplcc/Company/Company-List.html (last reviewed on November 28, 2018).