



Not Every (Open Source) Cloud Has A Silver Lining

By: James G. Gatto

This article was originally published on Law360.com on March 1, 2019

The use of open source software and cloud-based computing individually are growing at a rapid pace. The use of open source in cloud deployments is also growing. Until recently, the OSS license issues with cloud deployments have been fairly straightforward. It is well known that certain OSS licenses have some significant legal implications (detailed below) but that these implications are triggered when software programs that use OSS are distributed. Due to the fact that with most cloud-based deployments the software is not distributed, many developers are lulled into a false sense of security that there are no OSS implications with such deployments. The reality is there are a growing number of OSS licenses that have significant legal implications, even when the OSS is used in a cloud-based deployment. This article will address some of the relevant licenses and their legal implications.

Overview of Open Source Licenses and Their Legal Implications

OSS refers to a type of license, not a type of software. Absent any prior contractual obligations, the developer of a new software program can distribute that software under a proprietary license, an open source license or both (referred to as “dual licensing”).

Open source licensing is based on copyright law. In the U.S., copyright owners have the exclusive right to copy, modify and distribute their copyrighted works, among other rights. Proprietary software licenses typically prohibit copying (other than for back up purposes), modifying or redistributing the software. Usually, only the object code (i.e., the machine-readable code or “compiled form”) of proprietary software is distributed. The source code is not. Under most open source licenses, the recipient is permitted to copy, modify and redistribute the software.

Usually, the source code (*i.e.* the human-readable version of computer code that developers use to write a software program) of open source software is also made available (*i.e.* it is open). Practically, this is important because to modify software, a developer typically needs the source code. Legally, this is significant because even if a recipient can modify software, they need the legal right to do so. Open source licenses grant this right. Proprietary software licenses generally do not.

Open source licenses do not make the software to be deemed public domain. A developer of open source software retains copyright in the software. Public domain works, in contrast, are not protected by copyright (*e.g.*, either the copyright expired, the software has been dedicated to the public or otherwise ceased to exist, or because the software was not subject to copyright in the first place, such as with certain works created by the U.S. government). Public domain works can be used without restriction. Users of software covered by an open source license must comply with the terms of the license. Failure to do so can result in copyright infringement and/or breach of contract issues.

Hundreds of different open source licenses exist. Many of the licenses fall into one of two major types — restrictive licenses or permissive licenses. Restrictive licenses typically require that if any software includes or is derived from software covered by a restrictive license and such software is distributed, then the distributed software must be licensed under the terms of that restrictive license. Restrictive licenses keep software and its derivatives subject to open source license terms. Permissive licenses allow recipients to modify software and distribute the modified software under any type of license, even a proprietary license.

One of the most significant implications of restrictive licenses arises if OSS subject to a restrictive license is combined with other software (*e.g.*, proprietary software). Under some licenses, if the combined work is distributed, then the source code for the combined work must be made available, *including the source code for the proprietary software*. Additionally, the recipients are granted the right to further copy, modify and redistribute the software without charge. This is often referred to as “tainting” the proprietary software.

If your business objective is to charge for your proprietary software (and to preclude your customers from copying, modifying or redistributing your proprietary software), this is problematic. Tainting of proprietary software is one of the most important legal implications of using OSS for companies that have proprietary software.

In the vast majority of licenses where this implication exists, it is triggered by distributing the combined software. Despite the critical implications that can arise from “distribution,” often this term is not defined in the license. Generally speaking, a distribution occurs when a copy of a software program is made and that copy is provided to a third party. Providing network access (*e.g.*, via a SaaS or cloud model) to a software program typically is not a distribution because the user does not get a copy of the software (or rights to copy, modify or redistribute it).

As a result, if an entity uses OSS in work implemented via a cloud deployment (where the work is not distributed), then this implication is not triggered in such licenses. This means that one can use OSS governed by such a restrictive license, combine the OSS with proprietary code, and not be required to make the source available nor grant others the right to further copy, modify and redistribute the software without charge.

However, as discussed in the next section, there are several open source licenses where tainting can arise, even when the software is only accessed via a network (*e.g.*, via a cloud-based deployment) and is not distributed. In these licenses, the “network access” of the combined software triggers the obligation to make source code available.

Open Source Licenses That Raise Implications With Cloud Deployments

One of the most well-known restrictive open source license which includes a “network access” provision that triggers tainting is the [GNU Affero GPL license](#), or the GNU AGPL.^[1] The AGPL is a version of the GPL license that is nearly identical to the GPL, but with an important exception. Unlike the GPL, which imposes the most critical conditions only upon distribution, the AGPL triggers these conditions when the covered software is accessed via a network. Thus, combining AGPL components with proprietary software accessed via a network can taint the proprietary software.

According to the GNU AGPL: “... in the case of software used on network servers ... the GNU General Public License permits making a modified version and letting the public access it on a server without ever releasing its source code to the public.” To close this perceived loophole, the GNU AGPL states that it: “requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.”

With the rise of huge public companies using OSS for free in cloud deployments, a number of OSS developers have changed their licenses to prevent this. For example, three significant open source developers recently changed their license terms to address this issue.

Redis, which is an open source, in-memory data structure store, used as a database, cache and message broker, uses a license referred to as the [Redis Source Available License Agreement](#). One part of the license grants the right to use the software or modifications, “only as part of Your Application, but not in connection with any Database Product that is distributed or otherwise made available by any third party.” The term database product is defined in the license to mean:

any of the following products or services: (a) database; (b) caching engine; (c) stream processing engine; (d) search engine; (e) indexing engine; (f) machine learning or deep learning or artificial intelligence serving engine; (g) a product or service exposing the Redis API; (h) a product or service exposing the Redis Modules API; or (i) a product or service exposing the Software API.

The Redis FAQs explains why the license is limited in this manner. It states:

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.

[1] The Affero General Public License, version 1 (AGPLv1), was published by Affero, Inc. in March 2002. The GNU Affero General Public License is published by the Free Software Foundation, Inc. These licenses are often referred to interchangeably, however, they are different licenses. As noted in the GNU Affero General Public, “An older license, called the Affero General Public License and published by Affero, was designed to accomplish similar goals. This is a different license, not a version of the Affero GPL.”

However, some cloud providers have repeatedly taken advantage of successful open source projects, without significant contributions to their communities. They repackage software that was not developed by them into competitive, proprietary service offerings and use their business leverage to reap substantial revenues from these open source projects.

Redis Labs has been leading and financing the development of open source [Redis](#) for years, and at the same time believe we deserve the fruits of these efforts. While the Redis core is and will always remain available under the open source BSD license, in order to keep our business and the Redis project sustainable, we've decided to license certain modules built by Redis Labs (e.g. RediSearch, RedisGraph, RedisJSON, RedisBloom, RedisML) with the [Redis Source Available License](#).

Interestingly, the FAQs also address why it didn't just use the GNU APL. The FAQ indicates that initially they did use the GNU AGPL but adds:

However, we later realized that AGPL did not prevent cloud providers from creating managed services using our code. Furthermore, we received requests from developers at large enterprises to move from AGPL to a more permissive license, because the use of AGPL was against their company policies.

MongoDB has become a very popular open source database. It recently published a new license called the [Server Side Public License](#), or SSPL. This license is modeled after the GPL v3 license, but includes a specific provision that addresses using the software as a service. The license states:

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.

Another interesting provision is the scope of the term Service Source Code, as defined in the license, this 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.

That's a pretty hefty list of things that need to be disclosed.

The MongoDB FAQ provides the rationale for this as follows:

The market is quickly moving to consume most software as a service. This is a time of incredible opportunity for open source projects, with the potential to foster a new wave of great open source server side software. The reality, however, is that once an open source project becomes interesting, it is too easy for large cloud vendors to capture all the value but contribute nothing back to the community. As an example, MongoDB has become one of the most popular databases in the industry. As a result, we have observed organizations, especially the international cloud vendors, begin to test the boundaries of the AGPL license.

Given this risk, small companies are unwilling to make that bet, so most software being written is closed source. We believe the open source approach leads to more valuable, robust and secure software, and it directly enables a stronger community and better products. The community needs a new open source license that builds on the spirit of the AGPL, but makes explicit the conditions for providing the software as a service.

We are issuing a new license to eliminate any confusion about the specific conditions of offering a publicly available MongoDB as a service. This change is also designed to make sure that companies who do run a publicly available MongoDB as a service, or any software subject to the SSPL, are giving back to the community. It should be noted that the new license maintains all of the same freedoms the community has always had with MongoDB under AGPL — they are free to use, review, modify, and redistribute the source code. The only changes are additional terms that make explicit the conditions for offering a publicly available MongoDB as a service.

One other license of note is the [Confluent Community License](#). This license includes an “Excluded Purpose” clause which is defined as “making available any software-as-a-service, platform-as-a-service, infrastructure-as-a-service or other similar online service that competes with Confluent products or services that provide the Software.” The license grant specifically carves out this excluded purpose from the license grant stating: “Licensee is not granted the right to, and Licensee shall not, exercise the License for an Excluded Purpose.”

According to the CCL FAQs: “Under the Confluent Community License, you can access the source code and modify or redistribute it; there is only one thing you cannot do, and that is use it to make a competing SaaS offering.”

Some of the other licenses that can raise issues include the Open Software License 3.0, the Honest Public License, the [European Union Public License](#), the [Apple Public Source License](#), the Academic Free License and several of the Creative Commons licenses, just to name a few.

It has always been important to analyze open source licenses and the trigger conditions that raise legal implications. The simple mantra “we don’t distribute the software” has never been a panacea. The rapid increase in cloud and SaaS deployments combined with the recent wave of new licenses designed to limit the free use of OSS in these ways has heightened the need to carefully scrutinize these licenses to avoid unwanted implications. When using OSS in network accessed (*e.g.*, cloud, SaaS) deployments, it is imperative to review each of the licenses that govern the open source components to ensure no unknown or undesirable consequences.

For further details, please contact:



James G. Gatto

Open Source Team Leader
202.747.1945
jgatto@sheppardmullin.com

Sheppard Mullin has a robust Open Source Team that advises clients on the full range of open source legal issues, including developing open source policies, advising on the approval and use of open source; M&A and finance transactions open source diligence, remediation and contractual provisions; patent issues with open source; distribution of software under open source licenses; contributions to open source projects and much more.