Software Licensing and Open Source Software

Presented to
The Dallas Bar Association Intellectual Property Section

IRENE KOSTURAKIS, ***Houston, Texas***
BMC Software, Inc.

IRENE KOSTURAKIS
BMC Software, Inc. • 2103 CityWest Boulevard • Houston, Texas 77042
irene_kosturakis@bmc.com

Irene Kosturakis is Area Vice President, Chief Intellectual Property Counsel at BMC Software, Inc., where she drafts and negotiates licensing agreements with third parties and for BMC's inter-company arrangements. She also performs as consultant on licensing transactions other attorneys who support BMC's sales and other transactions. Additionally, Irene is responsible for protecting BMC's innovation and therefore helps develop and manages the Company's patent and trademark portfolio. She is on the Company's cybersecurity incident response team, advising the team with regards to the handling and disclosures of security of and vulnerabilities in U.S. data. For thirteen years, Irene was an adjunct professor at South Texas College of Law for its International Business Transactions course, teaching the licensing aspects of the course.

DISCLAIMER
The views expressed in this article are entirely the author's and do not reflect the views of her employer.

TABLE OF CONTENTS

Software Licensing and Open Source Software

I. INTRODUCTION

Intellectual property is what ensues from the mind's intellectual endeavors. It can be an idea, a conception, a discovery, or invention. It can be a work of authorship such as a poem, a story, a book or something as practical as a refinery operations manual. Whatever form it takes, in a business context, intellectual property can be the most valuable asset of a business. This is most true in the information age we live in – an age in which many assets are intangible, and we rely on information to do our work and run our businesses.

Not unlike real property, which is protected by tangible fences, gates, and locks, which limit access to keep others out, intellectual property must be protected to maintain its value. The fences are government-granted legal rights to exclude others from the use and enjoyment of the intellectual property. These intellectual property rights include limited exclusivity and legal remedies for the owner of the intellectual property for the violation of such rights. The owners are the creators, inventors, discoverers, authors, and thinkers of the intellectual property or their assignees.

Although sometimes misused interchangeably, there is a clear distinction between *intellectual property* and *intellectual property rights*; the two terms are *not* interchangeable. For example, one is the invention, another is the patent protection that is granted; one is the work of authorship, and the other is the copyright rights in such work. One is the idea, and the other is the trade secret protection that it has.

It is because of its intangible nature that intellectual property can be evanescent; it can easily slip through one's fingers, if nothing is done to protect it. For example, an idea shared with everyone benefits all who learn it and not just the idea generator. It is clear there is a relationship between the value of something and its exclusive enjoyment. i.e., having the ability to keep others from enjoying it. However, many times, the licensor of software doesn't want to keep others from enjoying their intellectual property. Instead, the licensor wants to allow others to enjoy it, but only on the terms the licensor specifies.

A license is the vehicle used to allow the licensor to specify the way licensees can exercise the intellectual property rights necessary to use the intellectual property. In law, a license is a "permission to make, produce, or use something that has been created by or belongs to another person or company, usually by paying for it." https://dictionary.cambridge.org/us/dictionary/english/license.

Companies rely on intellectual property protections in order to realize the value to their company for their research and development efforts, maintain a period of exclusivity for their intellectual property, and market themselves as innovative, and the companies have various ways of realizing this value. One way is to license the intellectual property. This article will focus on technology licensing. This article will look at software licensing, open source software licensing, cloud- or software as a service (Saas)-based considerations, and finish with some AI compliance tips.

II. INTELLECTUAL PROPERTY

A. Intellectual Property Distinguished From Intellectual Property Rights

Intellectual property (IP) is a legal concept that refers to creations, conceptions, and discoveries that ensue from someone's intellect. Under intellectual property law, if certain actions and precautions are taken, IP is entitled to protection, and its owners are granted certain exclusive rights. IP can be any of a variety of intangibles, such as discoveries and inventions; musical, literary, and artistic works; words, phrases, symbols, and designs used in commerce to indicate the source and quality of products or services; and information and ideas. Intellectual property rights protecting these intangible assets include copyright, patents, trademarks, industrial design rights, trade dress, and trade secret rights.

B. Intellectual Property Rights – Historical Development

Our country's founding fathers wisely recognized that stimulating the creation of intellectual property was required and important to the progress of this country's society. The authors of the Constitution, some of which were inventors in their own right, understood that a framework of laws was needed to protect those who would make the initial investment in research and development critically needed for a new country to advance. They provided Article I, Section 8 of the US Constitution as the basis of legal protection for IP in the US.

> The Congress shall have the power …To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.

Congress subsequently granted limited exclusive intellectual property rights in the form of patent and copyright rights to discoverers of inventions and authors of creative works, respectively. The end goal was to provide an incentive for inventors and authors to create and disclose their works, thereby building the country's IP and making the knowledge available for the public's benefit, mutually benefitting society and the IP owners. Fundamental to this goal is the recognition that invention yields improvement, which results in more invention and spurs the progress of society, but only if that invention, improvement, and progress are available to others to stimulate their imagination, provide the seed for invention, and fuel competition. If laws protecting the developer of IP did not exist, disclosure and sharing would be less likely because its creators would be concerned they would lose the commercial benefit of their creations if they could not exclude others from taking them.

This concern was addressed internationally almost a century after the writing of the US Constitution. In 1873, afraid their ideas would be stolen and exploited commercially in other countries, foreign exhibitors refused to attend the International Exhibition of Inventions in Vienna. *WIPO – A Brief History,* WORLD INTELLECTUAL PROPERTY ORGANIZATION, http://www.wipo.int/aboutwipo/en/history.html (last visited August 18, 2024). This signaled the need for international protection of intellectual property. In 1886, the Berne Convention for the Protection of Literary and Artistic Works (the Berne Convention) was enacted to protect works of authorship through copyrights. https://www.wipo.int/treaties/en/ip/berne/ (last visited August 18, 2024). In 1970, this organization became the World Intellectual Property Organization (WIPO), located in Geneva, currently having an impressive 193 member states, as diverse as the Holy See, and a mission and mandate to administer intellectual property matters recognized by the member States of the United Nations. https://www.wipo.int/members/en/. (Last visited August 18, 2024).

In the US, pursuant to the Constitutional license of Article I, Section 8, Congress enacted a system of centralized and federalized IP laws with the primary goal of promoting the progress of society. If not protected, IP has little to no value. IP's value arises from the exclusive right to use it, or, more importantly, from its corollary, the right to exclude others from using it. Exclusive rights granted under IP laws allow creators of IP to benefit from their discoveries and creations. In 2019, the US Patent and Trademark Office reported that IP-intensive industries create an estimated 63 million jobs for Americans, accounts to 41 percent of domestic economic activity, and accounts for $7.8 trillion in gross domestic product. https://www.uspto.gov/sites/default/files/documents/uspto-ip-us-economy-third-edition.pdf. (Last visited August 18, 2024).

There is a positive correlation between strengthening an IP system and economic growth. Economists have estimated that between one-half and two-thirds of the value of corporate market values in the US can be traced to intangible assets. BARUCH LEV, *Remarks on the Measurement, Valuation, and Reporting of Intangible Assets*, FRBNY ECONOMIC POLICY REVIEW (Sept. 2003) p. 1 http://people.stern.nyu.edu/adamodar/pdfiles/articles/LevonIntangibles.pdf (last visited August 18, 2024). While some may debate the amount of the value of IP to the US economy, most can agree that in countries with developed or developing IP systems, those systems most often than not result in an increase of intellectual capital, and competition is fostered.

III. TYPES OF INTELLECTUAL PROPERTY RIGHT LICENSING

A. Intellectual Property Rights Licensing

IP rights include patent rights, industrial design rights and design patent rights, copyright rights, and rights in trademarks, service marks, trade dress, and trade secret rights. There are other exclusive rights, such as circuit design rights protected by mask work rights, found in the copyright statute. 17 U.S.C.A. §901 et seq. (O'Connor's Fed'l Intellectual Prop. Codes Plus (2016-17)) [*United States Code Annotated*]. Each of these protects a distinct piece of IP by exclusively granting certain exclusive rights to that IP for a certain period of time or term of exclusivity, provided that substantive requirements for receiving such exclusivity are met.

This article focuses on copyright rights of software. As an information society, many new technologies and content (including music, video, writings, and software) comes protected by copyright rights. Copyright rights can be best be protected by their owners by using good practices in licensing.

B. Software Licensing and Open Source

Software transactions are essentially licenses. *Modern Licensing Law*, Nimmer, R., Dodd, J. 2017-18 Edition, Sec. 2:67, p. 229. Software is neither a good nor a service. Id. at 228. Therefore, it is neither covered under U.C.C. Article 9, nor under Article 2. Id. In fact, Article 9 categorizes software as an intangible. Id. However, software could be considered information. *Specht v. Netscape Comm'ns Corp.*, 306 F.3d 17 (2d Cir. 2002).

The consideration of software licensing is from two viewpoints: one is the license of a party's software out to another party, and the other is the licensing in of third-party software. These licenses may be a fully negotiated contracts, but oftentimes are merely an end user license agreement, found online, having non-negotiable terms, and with assent manifested through a click acceptance by the user ("click-wrap" end user license agreement ("EULA")).

There are three types of software licenses: (1) a license whose terms are non-negotiable, either in a click-wrap EULA format or a software license agreement in which the licensor has non-negotiable terms because the licensor has much greater leverage than the licensee, for example negotiating a license for software from Microsoft or for some of the other "hyper-scalers," such as AWS, Cisco, Oracle, or Google; (2) for bespoke software, whose terms are usually negotiable; and (3) for open source software freely available online but whose terms are non-negotiable. This portion of the article will be limited to relevant aspects of each type of software license agreement.

1. End User License Agreements ("EULAs")

The most important aspect of a EULA is the question of how to have an enforceable assent, i.e., how to show that the end user showed her assent to the terms in a EULA. First, the terms must be provided to the end user – all the terms, not just some – such that the user can read them, consider them, and show her assent to them. See generally, *Specht v. Netscape,* 306 F.3d 17 (holding that downloading software didn't indicate assent to an arbitration clause because the terms were not presented to the end user other than by having to click a link on another online screen). The end user must scroll through the terms before offered the opportunity to indicate her assent to the terms presented by clicking. Finally, the licensor, must be able to prove that all the terms were presented, viewable, viewed, and assented to. Some click-wrap agreements force the end user to scroll all the way to the end before the end user can click her agreement. This is the best way for the licensor to have a proper enforceable assent. Not forcing the end user to scroll through all the terms allows the argument that the end user did not see all the terms and can result in failed assent as in *Specht v. Netscape*. A related technical issue for the licensor is to have technology that can collect and store the fact of each end user's assent to the EULA's terms, such that the fact can later be proven.

2. Reliance Solely on Copyright Rights

Why must the licensor offer a license agreement? Instead, why not just rely on copyright law? The reason is that while copyright provides restrictions on reproduction of the software, distribution of copies of the software, and on the creation of derivative works -- all rights granted only to the author of the software -- there are many other protections the licensor needs. A license agreement gives the opportunity to the licensor to limit its liability, disclaim warranties, and restrict the licensee's actions regarding the software and how it may be used. The warranties typically disclaimed are the warranties of merchantability, fitness for a particular purpose, use for an intended purpose, freedom from defects, and freedom from any claims of infringement of intellectual property rights of a third party. The licensor also uses the license agreement to control the use of the software, such as the grant being for a single user for personal use, on a single computer, for a number of "seats" or users of the software, or for limiting the use to a certain field of technology or duration of time. Additionally, the licensor may be interested in restricting the transferability of the license granted. Certainly, the licensor will want to limit the licensee's reverse engineering or disassembly of the software's object code to prohibit the discovery of the software's source code, thus protecting it as a trade secret. Ultimately, the licensor's goal is to put the end user in a box with regards to the use of the software. That box can be rigid and non-negotiable if the software product is a low-end consumer product, such as Microsoft Office or Quicken, or for more sophisticated transactions, may be negotiated till there is mutual agreement.

3. Negotiated End User License Agreements

While licensing terms are rigid at the lower levels of license cost and sophistication, even enterprise software license terms can be a "take it or leave it" situation. You need high dollar values before the full set of license terms are on the table for negotiation. All software vendors would desire to have their terms accepted without modification by the licensee. As with all negotiations, however, it is all about leverage and which party has more of it.

Most software licenses are for a particular version and release of the software. Versions address new enhancements of the software. Licensees will be charged for a new version as a new product. Releases, on the other hand, address updates to the software to correct errors, fix bugs, and remedy vulnerabilities in the software. There is no software that is bug- or error-free. All software contains errors. Therefore, software developers/licensors will correct such errors and release them as releases to their installed base of licensees.

When negotiating a software license, one important aspect is to identify specifically what software is being licensed, what version, and even what release of the software. In some licenses, it may even be necessary to attach as an annex or exhibit of the software license a listing of the software components or modules. The goal for both parties is to be able to fully identify what it is that is being licensed. As mentioned above, the provision of updates and error corrections can be addressed by the addition to the definition of the software being licensed, a phrase such as "and all updates to and releases for the software."

Most software is very lightly warranted, if at all, and generally, the most that a software vendor will warrant the software is for conformance to its specification as described in the software documentation. Therefore, software vendors will not promise the performance of the software to be error free. Instead, the software vendor will sell a support and maintenance service to provide to licensees, in the form of future releases, error corrections, bug fixes, and updates to the software. Except for the price, this is optimal for the licensee because the licensor has more knowledge and expertise (and, if it has other licensees, the greatest interest) in the software to correct errors in it. This is also optimal for the licensor because it is the party who likely does and should have access to the software's source code to make modifications needed to fix errors in it. Only in bespoke software development agreements would a licensee have a chance of having a software developer vendor warrant the software to be fit for its intended use.

As for infringement of IP rights, usually there will be no warranty of non-infringement, however, software vendors will sometimes indemnify licensees for the software's infringement of third-party IP. An interesting twist to this has developed in the artificial intelligence technology area. Recognizing that its large language model (LLM) contains as much content, data, and open source software, as it could possibly scrape from online sites and addressing the copyright infringement concerns of end users, for the Microsoft Copilot software and Microsoft GitHub Copilot code-feeding software, the company provides what it labels as the Customer Copyright Commitment (CCC). The CCC, however, does not warrant that Copilot will not infringe third-party copyrights, and it does not indemnify end users. Instead, Microsoft's CCC states clearly that it will only *defend* end users from copyright infringement claims and that only if end users employ all the mitigations that are listed in the enterprise license agreement.

4. Open Source License Agreements

When on-premises software (as opposed to Saas software) is provided by a vendor for a fee or royalty, it is usually provided in object code format. Object code is what computers can perceive, read, and operate. In actuality, the object code is compiled from human-readable source code, i.e., source code that has been written by humans in some computer language, e.g., C++, Python, Java, or other language. Once compiled into object code, source code cannot be perceived by the end user, and in fact, this keeps the source code a trade secret. The default rule is that the source code of software is not disclosed and is retained as a trade secret.

Open source software is different in that its source code is open and available online and therefore disclosed to everyone. Anyone can take and use open source software. Open source software is now a mainstay in almost all software programs. No developer would re-develop code, programs, and software pieces when she can "freely" download them and put into the product she is developing. Some open source software pieces are so well known and ubiquitous that to redevelop them would be inefficient and akin to "reinventing the wheel." To redevelop such software could also be prohibitively expensive. Software developers will typically look for pieces of open source code available online, and they will typically have their personal list of favored pieces that accomplish necessary functions.

The reason for offering the source code openly online is for other developers (the open source community) to access it, improve it, build upon it, and make it ubiquitous by distributing, embedding, or combining it with other code. The philosophy is to leverage the diversity of a crowd-sourcing effort and thus encourage more improvement and innovation. The Open Source Initiative (OSI), a global non-profit, works to "helps build a world where the freedoms and opportunities of open source software can be enjoyed by all. The OSI supports institutions and individuals working together to create communities of practice in which the healthy open source ecosystem thrives." https://opensource.org/about. (Last accessed August 18, 2024).

Above, "freely" was purposefully placed in quotes because while open source software is freely available, that doesn't mean it is free of cost or requirements. *Why Open Source Licenses with a Commons Clause May Become Less Common*, Nicholas D. Petrella, Stephen E. Kabakoff, https://law-journals-books.vlex.com/vid/why-open-source-licenses-843657555 (last visited August 18, 2024). It might not be free of cost. Because the source code is freely available for the use, monetizing open source software has been relegated to charging to provide support services or consulting. Id. An example of that is Red Hat software. Furthermore, open source software is not free of restrictions on use; each piece of open source software comes with a its own license agreement. As a condition of use, the licensee must comply with the open source software's license terms and conditions. Most often, these terms disclaim all warranties and include the requirements that the licensee provide the license terms to the intended end users, who must also agree to such terms and conditions, and that the copyright notice embedded in the software be included in all copies of the open source software.

Many licenses approved by the OSI have become well-known and are often used to offer open source software under "permissive" open source terms, such as the MIT license, the BSD license, or the Apache 2.0 license. Such licenses allow licensees to modify the software and create derivative works that may be offered by the licensee under

different license terms. Id. Less typically, a few software developers create their own licenses using highly personalized terms, such as the requirement that the user make a charitable donation to the developer's favorite organization.

However, a category of open source licenses has much more impactful and potentially detrimental terms and conditions. Some open source software licenses require that as a condition of combining the source code with other code, the entire software must be offered under the license terms and conditions of the original source code. This means that in order to comply with the license, the entire work must become open source software, and the source code of the entire work must be made open and available. This type of license is sometimes called "copyleft" licensing. The GNU General Public License version 2.0 or 3.0 (GPL) impose such requirements. See e.g., https://opensource.org/licenses/GPL-2.0 (last visited August 18, 2019). This type of licensing is of particular concern to proprietary software vendors. Inadvertently combining a copyleft piece of code into their software could result in having to make their proprietary code open source.

Open source software has become so ubiquitous that in a nested hierarchy, many open source software pieces contain other open source software pieces. All the terms and conditions of all the open source software pieces must be complied with, resulting in a challenge only solved through open source software audits. Because of this concern, the prevalence of open source software licensed under copyleft terms and conditions, and the need to comply with open source license terms and conditions, any party developing software for provision as a software vendor should have a third-party software review/audit process. This might start with a code scan or audit by a company that specializes in this. The goal would be to audit open source software usage in proprietary software to give insight into security and license compliance risks.

A scan of the software gives visibility to what has been included in the software's code and can be performed periodically. The process should also include the requirement that software developers submit for approval the use of any piece of open source software, including the software name, version, release, and license terms and conditions. This will allow the software vendor's lawyers, contract managers, or compliance personnel to review the license terms, make sure they are being complied with, and restrict the combination of copyleft software with its proprietary code. The software version/release and terms and conditions of any software can be changed by its author(s), and open source software is not exception. The license associated with open source software can change at any time, so precise identification of what is being used is important, and any change to a different version/release should mean that the third-party software included in the vendor's software must be resubmitted for approval. The process can be streamlined by creating a database of versions of software licenses that are approved for future use so as not to re-review past approved pieces of software.

Likewise, any purchase or acquisition of software in a merger or acquisition, must be reviewed under the same process as part of the due diligence. Any piece of software that will ultimately be provided to end users should undergo this type of clearance. However, this brings us to Saas- or cloud-based software, i.e., software that is not provided to the end user. For accounting, efficiency (both for the end user and the licensor), Saas software is the preferred mode of providing software functionality to end users. The model is a subscription model, a pay-as-you-go approach to software licensing. Issues of security can be handled by proper segregation of tenants on the Saas platform. A request that can arise is the licensee's desire to perform a penetration test of the platform to discover vulnerabilities as required by its auditors. In order to protect the disclosure of damaging information, such requests must be responded to by the licensor requiring the protection of the results of such penetration tests to be disclosed only to the licensor and no one else, even the licensee, in exchange for the licensor agreeing to promptly remedy any serious deficiencies that are discovered by the penetration test. As for copyleft software being used on a Saas platform, the issue that will develop is for any open source software component used on the platform, which is licensed under for example, the Affero General Public License (AGPL). The development and use of this copyleft license plugged a hole in the family of GPL licenses, whose terms only apply upon re-distribution, which would never happen on a Saas software. *Why Open Source Licenses with a Commons Clause May Become Less Common*, Nicholas D. Petrella, Stephen E. Kabakoff, https://law-journals-books.vlex.com/vid/why-open-source-licenses-843657555 (last visited August 18, 2024). The use of AGPL-licensed software in a Saas platform's software requires that the entire platform's source code be opened and made open source software.

IV. CONCLUSION

Intellectual property is important to clients, whether they are businesses, entities, or individuals. Its value can be realized and maintained through licensing. Though protected by the exclusive rights granted under copyright, allowing third parties to use software is best accomplished by use of proper license terms. Open source software is ubiquitous, and its use in proprietary software must be discovered and must comply with the open source software's license terms and conditions.