Jus Corpus Law Journal

Open Access Law Journal – Copyright © 2023 – ISSN 2582-7820
Editor-in-Chief – Prof. (Dr.) Rhishikesh Dave; Publisher – Ayush Pandey

Open-Source Software and Intellectual Property Rights

Aayush Garg[a] Dr. Nisha[b]

[a]UILS, Chandigarh University, Chandigarh, India [b]Assistant Professor, UILS, Chandigarh University, Chandigarh, India

Received 24 November 2023; *Accepted* 22 December 2023; *Published* 26 December 2023

In the intricate realm where Open Source Software (OSS) converges with Intellectual Property Rights (IPR), this research delves deep into the nuanced dynamics, uncovering the challenges and opportunities woven into their intersection. The investigation meticulously explores licensing models, enforcement mechanisms, and the global landscape, exposing intricacies such as licensing uncertainties, heightened cybersecurity concerns, and community discord. Drawing insights from a comprehensive synthesis of literature, empirical data, and legal analyses, the study emphasizes the delicate equilibrium required to navigate the coexistence of OSS and IPR effectively. It underscores the pressing need for transparent licensing frameworks, robust enforcement tools, and proactive community engagement. Given the expansive nature of open-source collaboration on a global scale, adapting to diverse legal contexts emerges as a pivotal consideration. Looking forward, the abstract envisions a future where enlightened practices, educational initiatives, and technological adaptations converge to cultivate an environment conducive to innovation within the open-source paradigm. This visionary perspective anticipates a seamless integration of OSS and IPR, fostering a collaborative landscape that addresses challenges with strategic foresight and sustained progress.

Keywords: *open source software, intellectual property rights, licensing models, source code, software development.*

INTRODUCTION

Open-source software (OSS) is created and distributed with its source code, which is the basis of computer software, allowing anyone to check, inspect, modify, improve and enhance it. It is commonly distributed under a copyleft or permissive license, allowing anybody to use, alter, and distribute the program provided they abide by the license's requirements. The concept of open source originated from a kind of ethical revolt among software developers, which they manifested through the release of computer systems' source code. Both academics and industry professionals gradually took up the cause, and this model finally snowballed into a rival regime in the field of software development.

An entirely new way of thinking about software development is represented by open-source software. Unlike the widely recognized 'conventional', 'proprietary' and 'closed' source code models, the open-source software model places a greater emphasis on the program's source code's unfettered accessibility. Open-source code development is also not lonely or closed group work, in contrast to proprietary software.[1]

The phrase 'intellectual property laws' refers to a group of legal theories that grant innovators ownership rights over their intangible works, such as trade secrets, copyrights, and patents. Authors have exclusive rights to their writings, visual works, and other creative works under copyright; patents cover goods, processes, and other practical inventions; and trade secret legislation protects confidential information with commercial value.[2]

The link between OSS and IPR is complicated and frequently contentious. Open-source software is predicated on cooperation and unrestricted sharing on the one hand. IPR, however, can limit how it is used and distributed. The relationship between the two provides the legal and economic issues involved, the potential benefits, and the risks of Open-Source Software for

[1] Vikrant Narayan Vasudeva, 'Open-source Software Paradigm and Intellectual Property Rights' (2012) 17 Journal of Intellectual Property Rights 511-520 <https://nopr.niscpr.res.in/bitstream/123456789/15019/1/JIPR%2017(6)%20511-520.pdf> accessed 15 November 2023

[2] ' Intellectual Property, Computer Software and the Open-source Movement' (*CRS Report for Congress*, 11 March 2004) <https://www.everycrsreport.com/reports/RL32268.html> accessed 15 November 2023

enterprises and individuals. The relationship also includes the latest developments in this field, such as the emergence of open-source patent pools and the use of IPR to promote innovation in the OSS community.

OPEN-SOURCE SOFTWARE

Programs whose source code is publicly accessible can be updated and shared per user needs. Open-source software has a base or source code often available for anyone to improve or modify for reuse and accessibility. Generally speaking, users never see the open-source code that makes up software. To make a program or application function, anyone can alter and manipulate a piece of software. Programs can be enhanced by including features or correcting malfunctioning areas by programmers with access to the source code.[3]

According to the Cambridge Dictionary, *"Open-source software or information can be obtained legally and for free from the internet, and can be used, shared or changed without paying or asking for special permission."*[4]

According to the Dictionary of Archives Terminology, *"computer code that is developed and refined through public collaboration and distributed without charge but with the requirement that modifications must be distributed at no charge to promote further development"*[5]

According to the Collins Dictionary, *"Open-source material is computer programming code or software that anyone is allowed to use or modify without asking permission from the company that developed it."*[6]

Anything that can be altered by everyone because its design or code is made available to everybody and is publicly accessible is referred to as 'open-source'. Generally speaking, 'open-

[3] 'What is 'Open Source'' (*The Economic Times*) <https://economictimes.indiatimes.com/definition/open-source> accessed 15 November 2023
[4] 'Open-source' (*Cambridge Dictionary*) <https://dictionary.cambridge.org/dictionary/english/open-source>accessed 15 November 2023
[5] 'Open-source software' (*Dictionary of Archives Terminology*) <https://dictionary.archivists.org/entry/open-source-software.html> accessed 15 November 2023
[6] 'open-source' (*Collins Dictionary*) <https://www.collinsdictionary.com/dictionary/english/open-source> accessed 15 November 2023

source' items are distributed without charge, encouraging collaboration and open communication amongst many users who may choose to contribute their own ideas to improve or alter the product. Open-source software is computer software in the form of source code that is freely available to the public under a copyright license that complies with a set of standard requirements known as the Open-source Definition.[7]

THE OPEN-SOURCE DEFINITION

Introduction: Access to the source code is only one aspect of open source. Open-source software distribution agreements have to meet the following requirements:

Free Redistribution: No party may be prevented under the license from selling or distributing the software as part of a distribution of software that combines programs from many sources. A royalty or other payment shall not be demanded by the license for such a transaction.

Source Code: The program needs to have source code and enable both source code and compiled form distribution. When a product is supplied without the source code, there should be a widely advertised way to access it for no more than the cost of a decent reproduction—ideally, this can be done for free over the Internet. The recommended format in which a programmer would alter the program must be the source code. Source code that has been purposefully obscured is prohibited. Forms in between, like the result of a translation or preprocessor, are not permitted.

Derived Works: The license needs to permit the distribution of derived works and modifications under the same conditions as the original software.

Integrity of The Author's Source Code: If the license permits the release of 'patch files' containing the source code for the purpose of altering the program at build time, then the license may prohibit the sharing of source code in modified form. The distribution of software created

[7] Sharanya Mukherjee and Bedapriya Lahiri, 'IPR Concerns and Legal Aspects of Open-source Software' (2015) 2(4) International Journal of Law and Legal Jurisprudence Studies <https://ijlljs.in/wp-content/uploads/2015/06/IPR-Concerns-and-Legal-Aspects-of-Open-Source-Software_S-Mkherjee-and-B-Lahiri.pdf> accessed 15 November 2023

using altered source code must be expressly permitted by the license. Derived works can need to use a different name or version number from the source program according to the licensing.

No Discrimination against Persons or Groups: The license cannot be used to discriminate against an individual or group of individuals.

No Discrimination Against Fields of Endeavour: The program's use in a particular industry cannot be prohibited by the license. It might not prevent the program from being utilized for genetic research or in a commercial setting, for instance.

Distribution of License: Anybody to whom the program is redistributed must be entitled to use it under the terms of the program, without those parties having to execute an extra license.

License Must Not Be Specific to a Product: The program's rights must not be contingent on its inclusion in a specific software distribution. All parties to whom the program is transferred should have the same rights as those granted in association with the original software distribution, provided that the program is removed from that distribution and utilized or distributed in accordance with the terms of the program's license.

License Must Not Restrict Other Software: Restrictions on other software that is distributed with licensed software cannot be included in the license. The license should not, for instance, mandate that any other software delivered on the same media be open-source.

License Must Be Technology-Neutral: No clause in the license may be based on a specific technology or interface design.[8] Developers have free access to OSS source code that contains tens of millions of lines. Among the most well-known instances of OSS are:

Linux: an operating system for computers that is based on UNIX, the popular proprietary operating system used in the 1970s and 1980s for both consumer and commercial computers.

[8] 'The Open Source Definition' (*Open-source Initiative*) <https://opensource.org/osd/> accessed 15 November 2023

Mozilla Firefox: a web browser that was initially created by the nonprofit Mozilla Foundation, which is committed to encouraging innovation, openness and engagement online.

OpenOffice: It is an application package with parts for making and manipulating databases, spreadsheets, presentations, graphics and documents.

Apache HTTP Server: A commonly used web server program has contributed significantly to the expansion of the internet.

THE RISE OF OPEN-SOURCE

The linked but distinct 'free software' movement gave rise to the open-source software movement. A free replacement for the AT&T-owned Unix operating system was promised in 1983 by Richard Stallman, a programmer at the MIT Artificial Intelligence Laboratory. Stallman called his proposed OS GNU, a recursive abbreviation for 'GNU's Not Unix'.[9]

The University of California, Berkeley and the Massachusetts Institute of Technology are the two academic institutions that gave rise to the open-source movement. The GNU General Public License (GPL) and the Berkeley Software Distribution (BSD) License, which are well known as the movement's poles, were ultimately formed by the open-source ideologies that developed at these two universities. Nearly twenty years later, the Mozilla Public License (MPL) serves as the third pillar supporting the open-source movement, and Netscape Corporation created it.[10]

A group got together in 1998 to talk about how to advance the concepts of open collaboration and sharing code. The organization chose the word 'open-source,' created by Christine Peterson, to define its goals because they were concerned that the name 'free software' would make its ideas less appealing to companies that wished to keep portions of their code proprietary.[11]

[9] 'The WIRED Guide to Open Source Software' (*WIRED*, 24 April 2019) <https://www.wired.com/story/wired-guide-open-source-software/> accessed 17 November 2023
[10] Vasudeva (n 1)
[11] The WIRED Guide to Open Source Software (n 10)

The term 'Open-source': The term 'open-source' was coined during a strategy meeting in Palo Alto, California on February 3, 1998, just after the Netscape source code release was announced. The idea that the publicity surrounding the Netscape announcement had produced a chance to inform and promote the benefits of an open development approach led to the formation of the strategy session.

The conference participants thought that the practical, business-case justifications for Netscape's code release demonstrated an effective approach to interact with prospective software users and developers and persuade them to contribute to the development of open-source software by being members of a vibrant community. Additionally, the conferees thought it would be helpful to have a single term to distinguish this approach from the politically and philosophically focused term 'free software'. The word 'open-source' which was first proposed by Christine Peterson, finally came to light during the brainstorming process for this new label.[12]

Intellectual Property Rights and Software: Intellectual property rights provide the basis of the software industry. Depending on what kind of ownership rights you have, the law provides several ways to protect them. Four types of property rights protect software: trade secrets, copyrights, patents and trademarks. Each provides a unique kind of legal defence. Trade secrets, copyrights, and patents all contribute to the technology's protection. Trademarks safeguard the names or symbols used to identify a product in the marketplace.

What is Intellectual Property?

Intellectual Property or IP includes works of literature, art, inventions, designs, names, symbols, and pictures used for commercial purposes.[13]

[12] 'History of the OSI' (*open-source initiative*, 19 September 2006) <https://opensource.org/history/> accessed 17 November 2023
[13] 'What is Intellectual Property?' (*WIPO*) <https://www.wipo.int/about-ip/en/> accessed 17 November 2023

According to the Merriam-Webster Dictionary: *"Property (such as a concept, idea, invention, or work) that derives from the effort of the mind or intellect, also a right or registration (such as a patent, trademark, trade secret, or copyright) relating to or protecting this property"*[14]

In India, IPRs are protected under the following Acts:

I. The Biological Diversity Act 2002.[15]
II. The Copyright Act 1957.[16]
III. The Design Act 2000.[17]
IV. The Geographical Indications of Goods (Registration and Protection) Act 1999.[18]
V. The Patents Act 1970.[19]
VI. The Protection of Plant Varieties and Farmers' Rights Act 2001.[20]
VII. The semiconductor Integrated circuits Layout design Act 2000.[21]
VIII. The Trade Marks Act 1999.[22]

Software: Software is a part of the more prominent information and communication technology (ICT) industry, including electronic components used in ICT products such as computer hardware, telecommunications, equipment, and services.[23] Software is a set of programs or instructions that tell a computer what to do. Contrary to hardware, which describes a computer's hardware, it is the reverse. Applications, scripts, and other programs that operate on a device are together referred to as 'software'. It might be considered the unpredictable component of a computer, whereas the hardware is the fixed element.

---

[14] 'intellectual property' (*Merriam-Webster Dictionary*) <https://www.merriam-webster.com/dictionary/intellectual%20property> accessed 17 November 2023
[15] Biological Diversity Act 2002
[16] The Copyright Act 1957
[17] The Designs Act 2000
[18] The Geographical Indications of Goods (Registration and Protection) Act 1999
[19] The Patents Act 1970
[20] The Protection of Plant Varieties and Farmers' Rights Act 2001
[21] The semiconductor Integrated circuits Layout design Act 2000
[22] The Trade Marks Act 1999
[23] Stanley Nollen, Intellectual Property in the Indian Software Industry: Past Role and Future Need' (2004) International Intellectual Property Institute <https://iipi.org/wp-content/uploads/2012/10/India-Software-Study-2004.pdf> accessed 17 November 2023

INTELLECTUAL PROPERTY PROTECTION FOR SOFTWARE

Software Intellectual Property Protection in Software product innovation is frequently shielded as a form of intellectual property, usually by using patents or copyrights. However, they serve as distinct tools for the purpose, both copyrights and patents are meant to prevent others from misusing an innovator's work. In general, copyrights safeguard an idea's appearance; they do not, however apply to the concepts that are contained in a specific work. The Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPs)[24] of the World Trade Organization (WTO) solidified the worldwide application of copyright protection for software goods. WTO members must recognize computer programs as literary works as defined by the Berne Convention, whether they are in source code or as objects, under Article 10 of the TRIPs agreement.

Therefore, Software code is usually covered by copyright protection after it has been written and recorded on a media. A person who owns a copyright may utilize it to prevent others from using, duplicating, selling, or distributing the work without permission.[25] Historically, mechanical patent law and copyright law could be used to safeguard written content. Computer programs are a type of written work that is mostly functional, making it challenging to classify them within the current IP protection framework.[26]

The current intellectual property framework, which reached a compromise and protects different computer program components independently. The conventional means of protecting software was through trade secret law, which can shield secrets used in or implemented by software. While the legislative tool of copyright protects software's literal expression, Software now enjoys broad doctrinal protection against patents, which fundamentally shield software's

[24] 'Agreement on Trade-Related Aspects of Intellectual Property Rights (unamended)' (*World Trade Organisation*) <https://www.wto.org/english/docs_e/legal_e/27-trips_01_e.htm> accessed 17 November 2023
[25] Shivani Mathur, 'India: IPR In Development of Software Technology And Industry' (*Mondaq*, 26 May 2020) <https://www.mondaq.com/india/patent/940286/ipr-in-development-of-software-technology-and-industry?> accessed 17 November 2023
[26] Christina M. Reger, 'Let's swap copyright for code: The computer software disclosure dichotomy' (2004) 24(2) Loyola of Los Angeles Entertainment Law Review 215 <https://digitalcommons.lmu.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1469&context=elr> accessed 17 November 2023

technical representation. The application of technology protection measures, licensing, moral rights, trademark protection, and design protection legislation is also observed. Developers are free to decide whether or not to use all available measures of security; it is not required for them to be used with every software.[27]

TRADE SECRET PROTECTION

Open-source refers to user-to-user sharing of software designs and components. The idea of trade secrets as a legal weapon is to safeguard software. Nonetheless, given the track record of open-source software, trade secret protection is a challenge for developers, suggesting that idea sharing in the software industry helps developers more. A few drawbacks of treating source code as trade secrets include inadequate security, dangers to user privacy and data security, and fewer incentives for developers. Conversely, the open-source software development (OSS) approach offers several advantages such as enhanced skills, user-driven innovation, and competitive improvisation. As a result, the open-source software (OSS) model is more socially efficient than the trade secret model, which gives developers the option of either paying for inefficient software development or keeping an eye on a dominating company to stop potential abuses.[28]

In the early days of computing technology, when software was supplied individually under strict contractual supervision, trade secret law was preferred. But when technology advanced, it was no longer suitable or sufficient. Software is still protected by trade secrets, although this is no longer its primary or only method. This divide is widened by open-source software, whose philosophy completely contradicts trade secret law, making its use in open-source software impractical.[29] The Free Software Foundation (FSF) sees distribution of code as trade secrets as a GPL violation. Open-source trade secrets have also come up in court. In *Red Hat v SCO*, Red Hat

[27] David S. Evans and Bernard J. Reddy, 'Government preferences for promoting open-source software: A solution in search of a problem' (2003) 9(2) Michigan Telecommunication Technology Law Review 313 <https://repository.law.umich.edu/mttlr/vol9/iss2/3/> accessed 15 November 2023
[28] Saipriya Balasubramanian, 'India: Open-source Software In The Shoes Of Intellectual Property' (*Mondaq*, 04 February 2013) <https://www.mondaq.com/india/trademark/219484/open-source-software-in-the-shoes-of-intellectual-property> accessed 15 November 2023
[29] Vasudeva (n 1)

requested a declaratory ruling, claiming that since Linux is freely available to the public, it cannot be protected by trade secrets and that Red Hat has not breached SCO's trade secrets.[30]

COPYRIGHT PROTECTION

Computer scientists, lawyers, and legislators first started to consider forms of intellectual property protection that might be used for programs created using a combination of these mechanisms when they realized that it was possible to author, adapt, reproduce, and systematically load programming instructions into a computing device using a combination of a writable medium and a high-level programming language—understandable to a multitude of human programmers and (after compilation) computing machines.[31]

The exclusive right to reproduce a work by default belongs to the original author or creator, subject to certain restrictions, under copyright law. Subject to certain restrictions, the owner of a copyright may sell or license others the ability to duplicate his creations. For this reason, copyright infringement results from breaking certain rules and conditions. A common phrase in open-source licenses is 'copyleft'. An author can utilize copyright laws to stop other people from making, sharing, or modifying copies of his work. In contrast to the term 'Copyright', 'Copyleft' permits authors to make copies of their works and distribute them, however, these copies are subject to specific license agreements. As a result, copyright rules allow the authors of open-source software (OSS) to release their code under an open-source license, typically the General Public License.[32]

Public access and the extent of copyright protection are inversely correlated. Determining the extent of copyright protection for computer software becomes necessary as a result. Advocates of open-source software contend that the proprietary software model limits the advantages for

[30] *Red Hat lnc v The SCO Group, No. 03-772, Plaintiffs Complaint, Count 11, 74-77 (D. Dei. 4 August 2003).*
[31] P McCoy Smith, 'Copyright, Contract, and Licensing in Open-source P in Amanda Brock' in Amanda Brock (ed), *Open-source Law, Policy and Practice* 72 (2nd edn, OUP 2022)
[32] *Ibid*

society. To correct what was thought to be an incorrect situation, the open-source regime implemented its own interpretation of software copyright protection.[33]

In the *SCO v IBM*[34] case, SCO argued that IBM had illegally integrated its proprietary UNIX code into the open-source Linux operating system, violating their copyright and trade secrets. As a result, SCO requires that Linux users obtain a license from them for certain portions of the Linux code.

The SCO Group for giving Linux access to UNIX code sued several businesses. In the end, the court found that 326 lines of code in the Linux kernel might have been illegal. It is clear from the aforementioned instance that copyright laws are being fully utilized in order to enforce an open-source license. Thus, it is evident that copyright rules are being employed by an open-source license to safeguard the uses of the author's work.

MORAL RIGHTS

When it comes to open-source software, moral rights are regarded as being very important.[35] The open-source software methodology mostly relies on reputation and attribution benefits rather than cash incentives.[36] Either legal restrictions or the rights themselves govern software moral rights.[37] Certain nations acknowledge moral rights, whereas others do not. Certain nations expressly refuse software writers' protection under moral rights laws, or allow writers to voluntarily forego moral rights in contracts; still, other nations restrict the moral rights that software writers can exercise. *Section 57 of The Indian Copyright Act*[38] acknowledges the rights of

[33] Vasudeva (n 1)
[34] *SCO Group, Inc. v International Business Machines* Corp. 879 F.3d 1062
[35] P Suri and Associates, *Open Source And The Law* (LexisNexis 2006) 59
[36] Vasudeva (n 1)
[37] Greg R. Vetter, 'The Collaborative Integrity of Open-Source Software' (2004) Utah Law Review 563 <https://www.law.uh.edu/faculty/gvetter/documents/Vetter.CollaborativeIntegrityOfOpenSourceSoftware_8.22.2004.pdf> accessed 19 November 2023
[38] The Copyright Act, 1957, s 57

paternity and integrity with regard to computer programs; however, *Section 52*[39] limits the integrity right with regard to program changes made for backup copies and interoperability.[40]

Open-source software may already have some latent alternative protection under legal systems that acknowledge moral rights; in these nations, breaking the conditions of the open-source license would be against the developer's moral rights. For instance, in exceptional cases, the author may still use Section 14 of the Copyright Act to restrict the modifier's actions under German law, even though the GPL allows for a broad scope of OSS release. This is because the author is legally entitled to prevent the work from being altered or impaired in any way.[41]

PATENT PROTECTION

The open-source community views software patents with suspicion. Supporters of open-source software argue that the existing incentive structure is unjustified[42], especially in light of the success of open-source software, and they want to completely review patent jurisprudence in the context of software programs.[43]

Software patents are a serious danger to the spread of open-source. At the very least, in the open-source community, the development of open-source software would effectively end if it were found to have violated a proprietary program, even slightly. The problems faced by the open-source community are compounded by the fact that, on the one hand, software patents encourage covert infringement of open-source goods, and, on the other hand, open-source programs are vulnerable to patent surveillance.

[39] Copyright Act 1957, s 52
[40] Vasudeva (n 1)
[41] A Metzger and T Jaeger, 'Open-source software and German copyright law' (2001) 32(1) International Review of Intellectual Property and Competition 52
[42] Kirk D. Rowe, 'Why pay for what's free?: Minimizing the patent threat to free and open-source software' (2008) 7(3) John Marshall Review of Intellectual Property Law 595 <https://repository.law.uic.edu/ripl/vol7/iss3/9/> accessed 19 November 2023
[43] Matthew D. Stein, 'Rethinking UCITA: Lessons from the open-source movement' (2006) 58(1) Maine Law Review 157 <https://digitalcommons.mainelaw.maine.edu/cgi/viewcontent.cgi?article=1346&context=mlr> accessed 19 November 2023

The Free and Open-source Software (OSS) community envisions a software development and release strategy that includes this behaviour. A license under such a model grants the licensee the ability to alter and redistribute the program. When software is delivered in this way, the recipients may receive a direct license from the original licensor or an identical sub-license from the distributor, depending on the conditions of the license. The content may be freely adapted under all free and open-source licenses. These modified content may also be freely distributed under the terms of this license. Certain agreements, like the GNU General Public License, require that any software adaptations they cover be given under the same 'copyleft' license. The aforementioned makes it clear that the free and open-source software (OSS) licensing model offers a less expensive way for potential licensees who do not want to pay for a license to use and distribute their products. The ability to get patents on processes included in the code is nullified by the release of source code. Because of these challenges and the time it takes to execute patent sharing, the OSS community practitioners do not like the patent type of protection.[44]

TRADEMARKS

The term 'open-source' cannot be trademarked because it is descriptive. However, as consumers want a trustworthy means of determining whether the software is indeed open-source, OSI has established the OSI Certified certification brand. A list of open-source licenses that meet the Open-source Definition and have been vetted by the public and authorized by OSI is kept up to date by the organization.[45] When certification marks are used instead of trademarks, the open-source methodology is recognized rather than a specific OSS product. This also eliminates a number of trademark law annoyances, such as monitoring the mark and guaranteeing correct attribution, while permitting more flexibility in usage and assigning equal accountability to each developer.[46]

---

[44] Smith (n 32)
[45] Vivek Kumar Verma, 'Open-source Software (OSS) – Are they protected?' (*Indian Case Law*, 24 August 2014) <https://indiancaselaw.in/protection-of-open-source-software/#:~:text=Trade%20Mark%20Protection%3A%20Since%20the,a%20certification%20mark%2C%20OSI%20Certified.> accessed 19 November 2023
[46] Vasudeva (n 1)

OSS LICENSING

The development of free and open-source software (FOSS) is facilitated by open-source licenses. Laws pertaining to intellectual property (IP) limit how creative works can be altered and shared. These preexisting legal frameworks are used in the opposite way by free and open-source software licenses, which give liberties that encourage sharing and cooperation. They provide the receiver permission to use the program, view the source code, make changes to it, and share those changes. These licenses apply to computer programs for which creating modifications may require the source code. They also include scenarios in which the executable program sent to end users and its source code are identical. Permissive and copyleft are the two main types of open-source licenses. The source of permissive licenses is academics. Under some restrictions, they grant the freedom to disseminate and alter. A disclaimer of warranties and attribution to the original authors are typically required by these academic licenses. The free software movement is the source of copyleft licensing. In addition, copyleft gives distribution and modification rights, demands attribution, and denies warranties. The distinction is that reciprocity is required by copyleft. The source code for any derivative works must be distributed under a copyleft license.

These licenses provide the free use, modification, and sharing of software in accordance with the Open-source Definition. A license must go through the Open-source Initiative's license review procedure in order to be accepted by the organization, commonly referred to as the OSI.[47]

Different open-source licenses:

- A copyleft license is a sort of license whereby code that is generated from the original source code is granted the license terms.
- A license type known as a permissive license allows for greater flexibility in terms of distribution, modification, and reuse.[48]

[47] 'OSI Approved Licenses' (*open-source initiative*) <https://opensource.org/licenses/> accessed 19 November 2023

[48] 'Open-source Licenses: Types and Comparison' (*Synk*) <https://snyk.io/learn/open-source-licenses/> accessed 19 November 2023

Copyleft Licenses: Under this type of open-source software license, works may be freely distributed and modified, provided that any derivative works are likewise licensed under the same license. Its purpose is to guarantee the work's continued accessibility and openness to usage, modification, and dissemination in the future. Stated differently, your changes must continue to be equally available and open to all parties, with the same rights and benefits as the original.

These licenses for open-source software can be further categorized as follows:

1. Strong Copyleft licenses: A strong copyleft license must be used to safeguard source code, and the derivative program must also be made publicly available under that license. This covers all of the software's linked libraries and parts. For instance, if someone alters software and publishes it under a GPL license, they also have to provide the source code and grant a GPL license for their changes. Therefore, you should stay away from these open-source licenses if you want to develop software and sell it as proprietary.

2. Weak Copyleft licenses: While they only cover a small number of codes, the conditions of a weak copyleft license are comparable to those of a strong one. The only requirement of this open-source license is that the original or modified work's source code be made publicly available; any additional code used in conjunction with the work need not be published under the same license.

Permissive Licenses: One kind of open-source license that permits more freedom in the usage and distribution of open-source software is the permissive license. Software use, modification, and distribution are subject to a few limitations under permissive licenses like the BSD and MIT licenses. Permissive licenses give users the freedom to use the program for any reason, even for profit, and do not impose any restrictions on their ability to modify it or distribute the source code. This gives users more freedom in how they use and share the program, but it can also result in the development of exclusive, closed derivative works. Since permissive licenses provide users more freedom to utilize the software in their own projects, corporations and

individuals who wish to incorporate open-source software into their proprietary software frequently prefer them.

Comparing Open-Source Licenses: Key Distinctions: There are many different types of open-source licenses, and the Open-source Initiative (OSI) is important in this regard. A nonprofit organization creates and upholds open-source licenses to safeguard and advance open-source software.

PERMISSIVE OPEN-SOURCE LICENSES

MIT License: The Massachusetts Institute of Technology created the MIT license, which is among the most widely used and liberally worded licenses. Unlike many other licenses that are freely available, this one is concise and easy to understand. It says that the sole restriction on how the source code can be used is to maintain copyright and licensing information. However, since it doesn't clearly confer patent rights, some people would rather stay away from it. .NET, Rails, Babel, and other well-known open-source software products are all licensed under the MIT license.

BSD License: The Berkeley Software Distribution (BSD) license requests that the copyright notice and license language be included in the copy of the software, but it gives creators comparable rights to distribute the work under different licensing conditions and without source code. The most often used BSD license version is the 3-Clause license, while there are other variations as well. A non-endorsement clause in it forbids developers from endorsing their changes with the name of the original creators.

Apache License: The Apache Software Foundation (ASF) published the Apache license. Because it's a permissive license, developers can release their work under whatever license they choose as long as they give credit to the original creator and thoroughly record any changes they make. Some of your resources can be kept as closed sources, and you can get licenses to make money off of them or even get patents. Several well-known software solutions, like PDF.js, Swift, and Kubernetes, are licensed under the Apache license.

The Unlicense: This license is unconditional and public domain comparable, approved by OSI. The source code is completely free to use however developers see fit, with no restrictions. Those who simply want to release their software into the public domain without imposing any restrictions on its use—but who nevertheless need to include a license for legal reasons—select this permissive license.[49]

COPYLEFT OPEN-SOURCE LICENSES

- The GNU General Public License (GPL), which is appropriate for business, patent, and private usage, retains copyright terms and license notifications. Programs that use GPL code are required to provide all of their source code under the same license. Therefore, all of your source code must be released under the same GPL license if you distribute your program and use GPL code (for example, by utilizing a GPL library). The GPL is an effective copyleft license because of this restriction.
- The Affero GPL (AGPL) adds just one clause, but for certain software, it adds a crucial one. There is a gap for software that is only made available over the network, or that is not formally 'distributed' since the GPL license only kicks in when it is distributed. This gap is closed by the remote network interaction clause in the AGPL license, which makes any program used via a network subject to the GPL license.
- Copyright protection and licensing notifications are among the terms that the Lesser General Public licensing (LGPL) offers, on par with the AGPL and GPL copyleft open-source licenses. The main distinction is that the distribution of the larger project is not necessary for smaller projects or objects accessed through larger licensed works. Furthermore, the restrictions under which the updated source must be supplied do not match those of the main code project.
- For business software, the Eclipse Public License (EPL) is frequently utilized. Software created with EPL can be merged and sublicensed with non-EPL and even proprietary code, as long as the non-EPL components are kept separate and exist as distinct modules

[49] 'Open-source Licenses - Definition, Types, and Comparison' (*Solution Hub*, 03 February 2023) <https://solutionshub.epam.com /blog/post/open-source- licenses-definition-types-and-comparison> accessed 20 November 2023

or objects. Under the provisions of the EPL license, modifications are permitted, but they must also be released.

- The least restrictive copyleft open-source software license is called the Mozilla Public License (MPL). As long as any code licensed under the MPL is retained in separate files and provided with the software, they make it simple to alter and utilize their code in closed-source and/or proprietary applications. The MPL mandates the preservation of copyright notices and encompasses patent grants as well.

CLASSIFICATION BASED ON FUNCTIONAL DIFFERENCES

- Expanding upon the normal reciprocity commitment are the highly restricted licenses (strong copyleft). In other words, even when modifications and derivative works are made, the terms of the licenses cannot be altered. The General Public License (GPL) is a well-known example of a license with strict reciprocity requirements.
- The source code distribution terms—also referred to as 'copyleft' licenses—must be maintained in order to comply with restrictive licenses. As such, the terms of the license are unchangeable and will not alter even in the event that the source code is improved or modified later. Still, if the source code is integrated with other source code to create an entirely new work, then the ordinary reciprocity obligation does not apply. Lesser General Public Licenses are a well-known example of licenses with standard reciprocity requirements (LGPL).
- Without any restrictions on the dissemination of the altered software (a derivative work), the permissive licenses provide free distribution, copying, modification, adaptation, and even combining of OSS source code with the licensee's own proprietary codes. The Berkeley Software Distribution License (BSD), MIT License, and Apache Software License are a few well-known examples of liberal open-source software licenses.

CLASSIFICATION BASED ON HISTORICAL ORIGINS

- The Free Software Foundation and Richard Stallman introduced the GNU licenses in the 1980s, and they have a strong political message. Though lawyers are hesitant to use GNU

licenses because of their ambiguous language and unclear ramifications, software developers are familiar with them. Furthermore, because of their discrepancies with other open-source licenses, GNU licenses have a lot of issues.

- Academic licenses originated at US universities, most notably at Berkeley University. Academic licenses have explicit wording and are liberal in nature, so compatibility issues are not covered.
- Community licenses are derived from a number of excellent free software initiatives. The Apache licenses and the Artistic licenses (which use the Perl programming language) are the most widely used community licenses. In actuality, artistic licenses add a lot of unclear verbiage that can be ideal for hackers but not always suitable for attorneys. Legally speaking, considering patents and trademarks, Apache licenses are typically more 'hard' than artistic licenses.
- Corporate licenses: When Netscape made its web browser's open-source code available in 1998, it introduced the first significant corporate license. In actuality, corporate licenses cover a wide range of formal concerns not covered by other open-source licenses, including patent, copyright, and trademark licensing.[50]

Conflicts between Open-source Software and Intellectual Property: The rights of intellectual property and open source standards may conflict.[51] Some observers that the provisions of the open-source license will extend to the entire program and supersede any potential intellectual property rights if open-source software is integrated into a proprietary application have raised concerns. Furthermore, using software that some consider to be open source may give rise to allegations of intellectual property infringement from a party not obligated by the provisions of an open-source license. Lastly, concerns about the legality and enforceability of open-source licenses have surfaced.

[50] Maryna Manteghi, 'Understanding Open-source and Free Software Licensing Mechanism: A Close Review of the Alternative Approach to Traditional Notions of Software Licensing' (2017) SSRN <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3082313> accessed 20 November 2023

[51] Stephen M. McJohn, The Paradoxes of Free Software' (2000) 9 George Mason Law Review 25

The Alleged 'Viral' Nature of Open Source Software: There have been instances where certain open-source licenses have been called 'viral' in nature.[52] Certain open-source licenses stipulate that the conditions of the license must be automatically applied to all copies of the program, regardless of any modifications.[53] Some observers in cases have raised concerns where a programmer, maybe unaware, adds some open-source code to a more extensive software package. Under these conditions, the program's open-source component may 'contaminate' the whole thing. Therefore, even though the publisher intended for the application to be proprietary, it may alternatively be distributed in accordance with the terms of the open-source license. Therefore, any intellectual property rights that the publisher hoped to assert would be superseded by the open source software component.

The specific terms of the open-source license that comes with a piece of software determine whether or not it has the potential to spread 'virally'. Some pundits have cited the GPL license from the Free Software Foundation as an example of a possibly viral license. Anyone who uses or modifies the program is required by the GPL to make the source code completely available to the public, free of any proprietary interest, upon further distributing the program or a modified version of it. Because of this restriction, intellectual property rights do not protect software created and shared under the GPL. Therefore, by permitting users to alter and redistribute the software, but mandating that any modifications be made available to everyone under the provisions of the GPL, the GPL preserves the open nature of the open-source code. All programs that are developed from an initial open-source project should also be considered open-source in order for the open-source software to continue to be available to the public. These licenses permit updates and modifications to the code to be taken private and handled as proprietary software, even while they mandate that the original program that is given with the license stay open source.[54]

---

[52] Christian H. Nadan, 'Open Source Licensing: Virus or Virtue?' (2001-02) 10 Texas Intellectual Property Law Journal 349

[53] Robert W. Gomulkiewicz, 'How Copyleft Uses License Rights to Succeed in the Open Source Software Revolution and the Implications for Article 2B' (1999) 36 Houston Law Review 179 <https://digitalcommons.law.uw.edu/cgi/viewcontent.cgi?article=1074&context=faculty-articles> accessed 20 November 2023

[54] *Supra note 2 at 12*

Third-Party Infringement Claims: Certain open-source agreements, like the General Public License (GPL), essentially bar users from claiming they own intellectual property in open-source software. However, this restriction is only applicable to those who have given their consent for these licenses. Thus, the enforcement of intellectual property rights by non-licensed parties against software users is not inherently prohibited. Due to this, even if one party distributes software that it has declared to be open source, another party may claim that the software violates intellectual property rights. Such claims give rise to possible disputes between the owner of intellectual property rights and those who think the program is public domain.

The Impact of Open-Source Software on Intellectual Property: Software that is released under public license, enabling anybody to view, alter, and distribute it without restriction, is known as open-source software. The domain of intellectual property has been greatly impacted by this, especially in the areas of software creation and distribution.

The old model of private software development is challenged by open-source software, which has a significant impact on intellectual property. Previously, software businesses would create their own software internally, keep the source code confidential, and charge clients for licensing. The ability to examine and alter the source code of open-source software, however, encourages greater innovation and teamwork.

Open-source software can also make it more challenging for businesses to secure their intellectual property, which is another effect on intellectual property. It may be harder to stop people from copying and spreading it if the source code is publicly available. Businesses that depend heavily on their software as a component of their business model may find this especially difficult. However, it also offer chances to safeguard intellectual property. To safeguard their intellectual property, businesses can utilize open-source licenses, which permit anybody to access and alter the source code. The protection of the intellectual property rights of the original developer can be ensured by some open-source licenses, which demand that any derivative works be released under the same license.

It has a complex and wide-ranging effect on intellectual property overall. It offers chances for cooperation and creativity, but it also puts established software development processes to the test and may make intellectual property protection more challenging. As a result, it's critical that businesses and organizations carefully weigh the effects of open-source software on their intellectual property rights and look into alternate methods of protecting it.[55]

CONCLUSION

The complex opportunities and difficulties that exist in the dynamic space where Intellectual Property Rights (IPR) and Open Source Software (OSS) converge. The integration of existing research, empirical data, and comprehensive analysis have yielded significant insights into the complexities of intellectual property rights management in the cooperative field of open-source development.

The difficulties found, which range from unclear licensing requirements to unresolved legal issues and community disputes; highlight how difficult it is to strike a balance between protection and transparency. Clearer norms and increased awareness are necessary to address licensing ambiguities, which are frequently caused by the wide range of open-source licenses. Compatibility problems and difficulties with enforcement highlight the necessity of a unified approach to licensing and strong systems to guarantee adherence.

The difficulties that have been revealed thus far offer chances for development and advancement. Collaboration can be improved through proactive community participation, improved educational programs, and more transparent licensing models. Maintaining the success and creativity of open source projects will depend critically on how well legal frameworks are adjusted to global differences and how well they embrace the changing technology landscape.

[55] 'Intellectual Property Challenges in the Digital Age' (*Global Intellectual Property Convention*, 01 August 2023) <https://www.globalipconvention.com/blog/intellectual-property-challenges-in-the-digital-age> accessed 20 November 2023

In summary, the cohabitation of IPR with OSS necessitates a careful balancing act that recognizes the difficulties and makes use of the advantages that come naturally to collaborative development. Proactive steps and knowledgeable practices will be crucial for negotiating the complexity and realizing the full potential of this paradigm-shifting software as open source continues to define the future of software.