OPEN SOURCE SOFTWARE AND INTELLECTUAL PROPERTY

Boris Nedelkovski
University of Library Studies and Information Technologies, Sofia, Bulgaria
PhD Candidate

***Summary**: This paper explores the connection between open source software (OSS) and intellectual property (IP), analysing different topics such as patents, copyright, licensing and legal disputed. It delves into different types of open source licenses, focusing on the permissive and copyleft licenses, and analyses their effect on software developers. Furthermore, this paper debates the adoption of open source software by companies and how it coexists with their intellectual property policies. Major legal cases such as SCO Group vs. IBM, Oracle America, Inc. v. Google LLC, and Jacobsen v. Katzer are analysed and examined with the goal of illustrating the major intellectual property issues in the open source community. The paper highlights the importance of contributor license agreements (CLAs), license compliance and legal education in guaranteeing effectiveness in management of open source projects and conflict resolution. Lastly, the content puts an accent on the importance of enforcement instruments and collaboration in the software community in the direction of promoting transparency, growth and efficiency of open source software.*

***Keywords**: Open source software, Intellectual property, Permissive licenses, Copyleft licenses, Companies, IP policies, Legal disputes, Contributor License Agreements (CLAs), License compliance and compatibility, Legal enforcement and education.*

1. Introduction

The intersection and integration of open source software (OSS) and intellectual property (IP) rights has become one of the major focuses and focal points in today's digital world. This merger is challenging traditional forms of ownership and fosters new collaborative methods in the sphere of technology, thus making the connection between open source software and intellectual property increasingly significant.

Software refers to a collection of programs and data that include set of instructions and rules that enable the functioning of computer devices through different operating systems (Schryen & Kadura). Simply, the software is instructing computers and electronic devices what to do, thus making them work. The software may be of open or closed character. What differentiates the open source software from a closed one is the availability of the code in open

source code software, making it possible for other contributors do use, modify or share the software. On the other hand, closed software does not offer this possibility. This type of software allows only certain individuals with authorization to access the code. The public must usually pay in order to get the access to these types of software. The main driving factor of closed source software is profit, which is obtained through selling licenses to users. Even though profit is not the primary purpose for open software, there exist companies that do make profit, usually through charging for their support and consulting services. Closed software is more intellectual property friendly, by preventing unauthorized modification and sharing of data, while open software may not provide the same level of protection, but it rather promotes innovation and public exposure.

Intellectual Property rights cover the rights given to a person or a company for their creations and/or inventions (World Intellectual Property Office (WIPO), 2020). These rights might be in the sphere of patents – for novel inventions, trademarks – for unique and distinctive logos or names, and copyrights for original works. The legal framework that safeguards creations is the basis for encouraging innovation by granting the creators exclusive rights to their work. This paper will explore the dynamic interplay between OSS and IP and delve into the challenges and opportunities that rise from the coexistence of these two phenomena.

Traditionally, patenting is the best way of protecting inventions and with that - encouraging innovation in science. However, the nature of open source software, marked by openness and collaboration of different parties is contradicting to the modus operandi of patent protection and represents a unique challenge to the traditional patent system. This paper examines how can patents be connected to open source software and the potential conflicts that may come up in terms of patent protection.

Copyright law is giving creators exclusive rights to their work, while open source licenses are giving users the opportunity to use, change, adjust and distribute the software. While copyright law still applies to open source software, the granted licenses are means of collaboration and sharing which is not aligning with the principles of exclusivity protection that copyright offers. Open source projects do rely on copyright law in instances where they have to enforce licensing terms. Businesses that employ open source software must strike a careful balance between promoting teamwork and safeguarding their intellectual property. This paper delves into different strategies that companies use in order to align their intellectual property policies with the open source phenomenon. We examine the ways of handling the fine line between promoting innovation through openness while protecting unique technologies that bring number of advantages.

We can get valuable insights in the matter through examining legal cases that have had a special influence in the relationship between intellectual property law and open source initiatives. Through looking at cases such as The SCO Group vs. IBM and Oracle America, Inc. v. Google LLC, we explore how courts have dealt with issues from the realm of licensing, copyright violations as well as how do they enforce IP rights in the quickly developing world of open source software. Connecting these two aspects demands a very detailed and careful approach as safeguarding intellectual property while working in collaborative projects might be complicated. This paper delves into the practical difficulties that may be encountered in enforcing the IP laws that are in place in order that the open source software complies with the legal standards. To recapitulate, this in-depth analysis of Open Software and Intellectual Property aims to present a comprehensive understanding of the intricate intersection between joined effort to create and develop and legal frameworks. By analysing patents, copyright, various company policies and different legal challenges, as well as different ways to enforce intellectual property rights, this paper tends to give the readers a frame that supports well informed decision making.

2. Open Source and Patents

Patents are legal means given to inventors and companies employing inventors by the government, which grant them exclusive rights to their own inventions for a specific time period (Majerus, 2006). In relation to the main topic of this paper – the software context – patents on software are protecting novel inventions in the realm of computer programs and algorithms by giving inventors the exclusive right to make, use, sell or license their patented software invention. Patents are granting exclusive rights to inventors, while the main goal of open source is sharing and collaboration. It is logical that from majority's perspective, there should not be any connection between patents and open source software, predominantly because of the nature of these two subject matters. Nonetheless, there are cases where patents and open source software overlap.

What makes open source projects distinctive is that they use licenses (such as the ones approved by the Open Source Initiative). Once contributors obtain these licenses, they can freely use, adjust, modify and share the software in question. The main points of this approach are collaborating and permissive approach that aids to preventing conflicts related to patents. Consequently, open source software is normally not patentable. These projects operate under licenses which license users to view, modify and distribute the source code freely, which encourages transparency in the software development.

In contrast to that, the 20-year patent protection granted by the government to patented inventions might appear as being contrary with the principles of open source and many argue that it can even obstruct innovation and be misused to constrain competition. Sometimes, event groups that support open source projects are obtaining patents in order to secure defence, which will give them protection from potential legal challenges or various disputes and oppositions from legal entities that already have obtained patents related to comparable technologies. This phenomenon has been seen as e measure of precaution that aids in operating freely. The pros when using an open source software are lowered costs linked to patent filing, prosecution and renewal. Ultimately, the decision between patenting the inventions and avoiding patenting and opting out to open sourcing always aligns with the goals of the developers, with open source prioritizing transparency, collaboration, community and enabling a diverse and active creative community over protection of exclusive rights.

3. Open Source and Copyright

Copyright law governs how contributors' rights and duties are handled and it has a great effect on how software is established and shared. Even though the connection between these two is complex, both copyright and open source software protect creators. On one hand, open source projects give the authors credit for their accomplishments and advancements in the field and this aligns with copyright law, whose primary goal is seeing authors as the rightful owners of their creative works (Smith P. M., 2022).

Major differences apply, specifically when discussing how software is shared and the rules guiding and regulating this process. Open source licenses are different from copyright law in that they let users' access, modify and share source code, while the usual rules of copyright law usually restrict and heavily regulate what you can do with someone else's work, forbidding the actions that are allowed and promoted through the open software projects. Open source licenses that, which are based on copyright law, and walking a line between giving users some form of freedom one side and on another – making sure that contributors get the credit that they deserve. Consequently, balancing these two rights is a crucial part of the correlation between open source and copyright law. This correlation is very crucial and a vital part of how developers are motivated and new ideas are promoted. This link between copyright law and open source software builds a culture of shared information which creates a special environment that guides how software is developed in a world where legal regulations and community driven ideas can collaborate in improving software technology.

Both phenomena protect creators, with open software projects ensuring giving recognition for contributors through appointing them as authors or contributors to their projects, which aligns with the principles of copyright law, whose fundamental idea is the appointment of the creator as the rightful owner of the work. This principle marks a shared respect for authors.

If we look at the sharing and licensing models, we will spot certain discrepancies between open source software and copyright. At the core of open code source is the license covered possibility to access the code, use it, but also modify it and distribute it. This possibility is heavily departing from the traditional copyright limitations that are omni present and reflects the unique open character of open source licensing which introduces a form of flexibility that transcends standard copyright allowances.

An important mutual factor between open source and copyright is the teamwork, with open source licenses creating an environment where smaller or bigger number of contributors work together on a project. Open source licenses operating under copyright law, allow different people to add to the project without any issues, they can all see and use the code, which proves that and open source community can successfully work together. Finally, the synchrony between open source and copyright law is a relation that protects the rights of creators and strengthens innovation and teamwork.

4. Open Source and Licensing

Licensing is the actions which involves the right holder giving another person the legal rights to use, modify or distribute their intellectual property (Starita, 2023). When it comes to software development, licensing is of great importance as it creates a legal frame around the way and the extent to which software is used, modified and shared.

The key distinctiveness of open source code software is that it freely shares its code with the public, but even so, there are specific licensing terms that must be followed when utilizing open source code which determine how users can use, adjust or distribute the software (Wikipedia, s.d.). There are a number of different open source software licenses, some with similar but some with very different terms that define the scope of actions within the software project. This paper will analyse the two most common types of open source software licenses.

Permissive Licenses:

We will start with analysing the more open, adjustable, flexible and accommodating type of licenses. Permissive licenses carry this name as they are known for not having many restrictions (Smith, 2022). The MIT License and the Apache License are in my opinion the two

most mention worthy examples when it comes to permissive licenses. The MIT license is very flexible as it gives the contributors the freedom to do almost everything with the software (Fossa, 2021). The only "must" is that they include the original copyright and license notice when sharing the software. It also states that the software is provided "as is", without any warranty. This license is one of the most popular ones within the open source community as it grants its users extensive freedoms to incorporate MIT licensed code into proprietary works without having to release their derivative works under the same terms.

Similar to the MIT license, the Apache license also is very lenient in its terms and gives freedom as long as users keep original copyrights and licenses notices. The owners of the software protected by Apache license, grant a patents license to users and in doing so they protect users from patent related issues. The Apache license makes it obligatory for users to keep copyright notices, disclaimers and attribution notices when sharing their modified version of the original software project. If the software is changed and shared under the Apache License, specific branding requirements must be followed. In both cases, these licenses aim to balance the freedom to use the software with the need for proper attribution and responsibility. Developers often choose between them based on the specific requirements and philosophy of their open source projects.

Copyleft Licenses:

Even though copyleft licenses are the foundation of open source communities and promote freedom, transparency and teamwork, they are still more restrictive that the permissive licenses (Smith, 2022). Copyleft licenses contain obligatory conditions which make contributors share any modifications or adjustments under the same terms. The main idea of this conditions is that the freedom granted by the original license are kept through the following distributions, which will not break the cycle of openness and collaboration.

A great example of a copyleft license is the GNU General Public License (GPL) (Heffan, 1997). This license makes it obligatory for the users to share the modifications that the users make under the same license, which would mean that any modification, adjustment or new work must also remain open source. With this decision, copyleft licenses make it possible to preserve the liberty and independence that comes with the open software in its future versions. Let's say company A releases a software under the GPL and company B modifies od adjusts it in any kind of way, company B then must share the modified source code with anyone that they share the software with or share it as open source with the receiving end of the software.

Two new versions of GPL software were introduced, GPLV2 which is the second version and GPLV3 which is the third version and was first introduced in 2007. These licenses are both accepted as having strong copyleft provisions. GPLV3 license is focusing on present-day matters such as compatibility with other licenses, patent provisions and protection against Digital Rights Management (DRM). Generally, GPL copyleft software license has played a key role in the promotion and growth of open source software phenomenon by inspiring and pushing teamwork, transparency and user independences. Additionally, this important license is providing software contributors the chance to keep their works open source which contributes to the bigger picture that consists of values of sharing.

5. Intellectual Property Policies in Companies Embracing Open Source

Intellectual property rules are implemented by companies in order for them to protect their intangible assets which may vary from software and patents to copyrights and trade secrets. These rules form a legal framework on how others interact with the intellectual property created by these companies or entities. The interest of companies to invest in open software projects is rapidly growing due to the fact that open source software projects are cost effective, they inspire collaboration and have big potential for innovation.

The rules of intellectual property are defining how companies collaborate with open source projects. It is common for technology companies to often use a mix of their own software and other open source software in order to get the best result which will lead to more success. Balanced approach in this area is crucial while taking this kind of approach that lets companies enjoy the advantages of open source software while protecting their exclusive assets.

Open source projects are often used for simple, non-specific components, which allow companies to put their main focus on their unique offerings. When it comes to licenses, this combination can be complicated as open source licenses have different rules which are crucial for companies to follow, check for rule changes and make sure that they will work and fit with their own parts. Companies must have a proper education on the extent to which and the way how licenses affect the sharing, modification and using open source code in their projects. This is the only way for them to stay compliant with the laws.

When using open software, there is a big chance of license issues and unintentional violations which can be prevented by effective risk management. Companies are more often than ever analysing and examining in depth different licenses, they do compliance checks as well as establish moderation policies. Having in mind the fact that open source software is the epitome of collaboration, sharing and innovation in the digital world, it gives amazing

opportunities to companies that mic proprietary works and open source software. They can benefit from the creativity of open source software communities while keeping the freedom to innovate on their own projects.

Companies often times use third party service providers. When doing so, clear communication is about open source dependencies, license details and compliance measures is key in order to avoid legal issues. It is crucial to direct the legal and regulatory aspects of intellectual property in a frequent manner in order to keep up with the rules and safeguard their ideas. Getting a good grasp of the legal regulations of using their own projects in combination with open source tools aids greatly to avoiding legal problems. Employees have to be taught about licenses, related rules and handling both simultaneously in an orderly manner. This will guarantee that every side is on the same page, while pushing forward smart use of open source applications.

Another aspect worth analysing is the intellectual property policies which represent an important aspect for most businesses. It is imperative that these policies are clear about the use of open source software in their technology (Smith & Mann, 2004). Companies must figure out and clearly point out where open source tools fit in with the goals of the company. The policies should also state that using open source is matching up with the overall plan of usage of technologies overall. IP policies of the company should explain how the companies use open source licenses by explaining what the licenses allow or don't allow as well as what kind of effect might they have on the company's own code. Distinguishing between which side owns what is key. Companies must state in advance what happens if persons employed by them are the contributors to open source projects. If this is the case – do these contributions belong to the company?! Policies should also state how the company will protect their own ideas while working in the domain of open source software. In summary, all affected sides should be taught about open source software and what kind of position will it take in the general picture of the company. This may involve related training programs that have the goal of keeping everyone up to date. As the open source practices change, so should the intellectual property policies of the companies. Flexibility in policies is crucial. Overall, solid IP policies help companies use open source software in the right manner, while still safeguarding their own ideas.

6. Open Software System & Intellectual Property Court Cases

The SCO Group vs. IBM

The legal contention between these two entities began back in 2003 when The SCO Group (formerly known as Caldera International) sued IBM, claiming that IMB violated the conditions written and signed in the contract about the work of the Unix operating system (SCO Group v. IBM, 2017). The main allegation was copyright infringement, with SCO claiming that IBM used SCO`s UNIX code in the Linux kernel. SCO claimed that IBM used a specific code that belonged to them and this is why the pursued significant damages from IBM.

This case was bigger than just SCO and IBM. It attracted major attention from all sides of the open source code world due to its potential impact on the Linux open-source community and the broader use of open-source software.

The legal proceedings of this case were initiated with SCO accusing IBM of underwriting exclusive to SCO UNIX code into open-source Linux operating system. SCO requested for damages to be paid back and sought injunction against IBM`s use of the code in question.

IBM counter sued, stating that this claim has no basis and it is only brought out in order to harm IBM`s business and reputation.

During this process, SCO had a hard time providing any clear evidence that supports their claim that they own the UNIX code, thus extensive and detailed examinations and analysis have been done throughout the legal process which would prove the validity of SCO`s allegations.

The court concluded that the Unix copyright was owned by another company – Novell and not SCO, which majorly weakened SCO`s case (The SCO Group, Inc. v. International Business Machines Corporation, 2017). Additionally, SCO files for bankruptcy in the same year which complicated the resolution of this case. The bankruptcy court approved the sale of SCO`s UNIX assets to UnXis in 2008 but they too could not pursue their copyright claim against IBM as in 2011 the court ruled that the Unix copyright was owned by Novell and not SCO.

In 2016, the U.S District Court in Utah concluded that SCO lacked the claimed copyright, which ended this long legal battle in favour of IBM.

This case represents an impactful battle in the technology industry which underlines the importance of software licensing agreements and intellectual property rights and the complexities that follow this relationship. The SCO Group used an aggressive rhetoric against

IBM, stating that IBM has infringed their copyright by illicitly contributing proprietary UNIX code to the open source Linux operating system. Their claim was based solely on their way of interpreting the licensing agreement and it was difficult to prove in court. Their existing legal claims were deemed not true, and that, combined with their inability to provide any further evidence has undermined their legal position and hurt the company's reputation.

This impactful court case illustrates the difficulties connected to intellectual property and software. It shows how challenging is to prove ownership and infringement when it comes to open source software. From this court case we can all learn how important is to have clear ownership documentation and to adhere to the licensing terms related to open source. When legal battles such as this one prolongs and take more than a decade to get resolved, they can harm the reputation of the company involved, thus it is crucial to have a well thought and strategic legal plan in intellectual property disagreements. Another lesson taken out from this court case is that contributors always keep the rights that belong to them as a result of their ideas. Additionally, it shows how companies can use counterclaims in legal battles and get into advantageous position due to them. In short, SCO v. IBM serves as a guide for the future which will help companies working with open source software better understand the legal side of their business.

SCO Group vs. IBM court case teaches us that no one can pursue a competitor in court with lack of compelling evidence, especially when challenging the validity of open source contributions. Aggressive litigation tactics are not always a good idea and might destroy company's reputation and credibility.

Oracle America, Inc. v. Google LLC

Oracle America, Inc. is an international tech company whose main point of development and managing is database software. This court battle is basing on the claims that Oracles copyrights was infringed from Google by Google using Java APIs in its Android operating system. Back in 2010, Oracle Inc. bought Javas rights and argued that Google did not ask for permission when copyright certain Java APIs. According to them, this was more than enough to constitute a blatant examples of copyrights infringement. On the other hand, Google claimed that APIs are crucial for innovation as well as software compatibility.

The legal proceedings were initiated when Oracle claimed that Google, directly copied and utilized certain Java APIs without the appropriate license, consequently labelling it as copyright violation.

Google argued in their defence that their actions fall in the domain of fair use and not copyright infringement (GOOGLE LLC v. ORACLE AMERICA, INC, 2021). Their actions were necessary for operating and innovation in general. They argued that they simply re-implemented Java APIs in order to create a new product what would be a reframed platform called Android.

During the first trial in 2012, the court confirmed that Google indeed infringed the copyrights of Oracle, but could not agree if this can be classified as fair use. Additionally, the judge ruled that APIs were not subjected to copyright protection.

During the appeal proceedings in 2014, Oracle appealed and the Appeals Court overturned the ruling and decided that APIs were in fact subject to copyright, which sent the case back to the trial regarding fair use.

The second trial took place 2 years later. The jury stated that Google`s use of Java APIs was fair, on which the judge agreed and sided with Google. The key reason for this was the decisions positive impact on innovation.

This court case proceeded through the years 2018 and 2021. Since Oracle continued to appeal, the Appeals court in 2018 reversed the fair use finding, but then in 2021, this court case was closed after the U.S Supreme Court confirmed the fair use of Java APIs.

One of the key takeaways from this case is the concept of fair use when it comes to copyright law. Googles held the position that their use of Java APIs was indeed a fair use, as it served a different purpose, in this case building a mobile operating system. The courts ruling has shown a different understanding of the term fair use under which what constitutes fair use can be defined in a slightly broader way.

The outcome of Oracle America, Inc. v. Google LLC serves as a precedent for future cases that will involve the use of APIs and the fair use phenomenon and it will serve as a guide for navigating similar legal issues.

This decision has given the green light for remodelling and remaking software interfaces in order to make various programs work better together. This was a major step for the world of software as it gives clear instructions and guidelines as to the copying of certain software parts and how you can utilize these parts without permission from the proprietor. Oracle America, Inc. v. Google LLC showed how complicated it can be to find out who owns what in the realm of software and stressed the importance of carefully examining legal guides when creating new programmes. We can all learn the importance of educating ourselves about intellectual property law and following the evolving regulations, even when working on open software projects. In

short, this court battle adds to the never-ending debate on how to protect new ideas whole still making sure that innovation is flourishing.

Much can be learned from this case, by both open source contributors and distributors. Contributors have the obligation to ensure that their contributions are created correctly, exactly as the license requests. On the other hand, distributors have to properly review and adhere to the terms of the open source licenses when it comes to distributing derivative works.

Jacobsen v. Katzer (2008)

This case is an interesting mix between open source software and the model train industry, thus receiving the nickname "Model Train Case". The lawsuit was filed by Robert Jacobsen, a model train enthusiast, against Matthew Katzer and Kamind Associates, accusing of copyright infringement and open source license breach.

Robert Jacobsen has released a computer program named "Decoder Pro" through an open source license that mandated attribution and the availability of the source code (Shagall, 2008). Matthew Katzer (through Kamind Associates) used the DecoderPro software, but did not obey the attribution requirement as well as source code availability requirements of the software license.

The so-called Artistic license under which Jacobsen released DecoderPro, is a permissive open-source license that allows free use, amendment and sharing of the software. Nevertheless, this license imposes conditions such as attribution requirements plus the need to provide access to the source code.

The U.S. Court of Appeals for the Federal Circuit made a significant decision, that confirmed that the terms in the open source licenses are lawfully enforceable under copyright law (Jacobsen v. Katzer, 2010). This decision ruled that the violation of open source license terms can indeed result in copyright infringement claim.

The court has ruled that the license terms were not merely contractual conditions but explicit conditions granting copyright permission, thus violating these conditions would result in the loss of the copyright permissions. This decision strengthened the idea that copyright law could be used to enforce open-source license terms, which was crucial step in the interconnection between intellectual property law and open source software. This was very important as it came in a time when some argued that violating open-source license terms should only bring contractual remedies, and not copyright infringement claims. This was especially effective towards users who could benefit from the open source software while avoiding to comply with the terms of the licenses.

The decision of this case has a great impact on the open source community and is welcomed by legal scholars as it brings legal certainty and ensures legitimacy of the license terms. Looking at the bigger picture, this decision is net positive for nurturing the open source culture and makes is easier for developers to release open source software when they know that the license terms will be obeyed.

The Jacobsen v. Katzer case has an important impact in establishing a legal framework that recognizes the importance of obeying the open source license terms as well as respecting the collaboration principles and respecting transparency. This ruling has had an effect on number of subsequent cases and remains cited in related discussions.

The case underlines the importance for individuals and organisations to review and comply with the terms and requirements outlined in open source licenses in order to avoid legal disputes. This was reaffirmed by the courts ruling in favour of Jacobsen and demonstrated that open source licenses, such as the Artistic License are legal instruments for protecting the rights of creators and contributor. The outcome of this case serves as a valuable example for both open source contributors and distributors. Finally, the case underscores the importance of comprehensive education when it comes to open-source licensing and intellectual property rights. In order to avoid complications and risk of legal disputes and ensure compliance with license terms, individuals and different entities must get proper education about the best practices for license compliance.

7. Navigating Legal Challenges in Open Source Adoption

In this section the paper will delve into dealing with numerous different legal aspects and possible difficulties in relation to using, contributing to or distributing open source software. Important points to reflect include:

Intellectual Property Issues

Abiding by intellectual property rules when developing open-source software is key to insure clarity, ownership and compliance. Some of the key areas of discussion will follow.

Ownership and Contributor Agreements have a key role in determining who owns the contributions and is important for avoiding disputed in collaborative efforts (European Union, 2023). Usually, individual contributors are the owners of their project and in order to respect the terms of the open source license, they grant specific user rights to the project. For larger projects, ownership of the copyright can be assigned by the body responsible for managing the project. Ownership agreements explicitly state the terms for contributors, ownership and

licensing rights making it easier to avoid disputes over intellectual property rights. They minimize legal risks and decrease legal disputes regarding ownership, licensing and contribution by formalizing the rights and responsibilities of contributors through and agreement. Contributor agreement clarifies under which terms the different sides contribute code, documentation and other important assets for the project. As open source projects more often than not, integrate code from various contributors, each potentially utilizing different license, contributor agreements make sure that the contributions line up with the correct open-source license. Other terms that are included in these agreements are terms in regards to licensing compatibility, code quality, patent grant etc.

License proliferation is the existence of large number of licenses, which makes it hard for developers to choose the right license (Gomulkiewicz, 2009). Firstly, compatibility confusion is created when the number of licenses is large and it is difficult to know if one software is compatible with another. Secondly, all of the available software has license terms that have to be red and understood and reading through a large number of different legal documents can be confusing. Thirdly, the community is highly divided as everyone is using different licenses which would be kind of like speaking different languages, which makes it harder to find a common ground. In order for the open source community to make the processes, collaboration and innovation simpler, it has to deal with license proliferation.

Addressing patent consideration is crucial for clarifying how patents coexist with the selected license. Open source licenses make sure that patents do not restrict the availability of the software by using clauses which grant the necessary patent rights to users so they can freely use, change or share the software in question. Other open softer licenses on the other hand, include patent grants in the license itself, through which users are automatically provided the patent right when using the software.

It is crucial to identify and attribute to the origin of the code since the code can originate from various sources. If we want to track the history of the code, we need to be able to find out the origin. This is crucial for correct contribution of creators as well as obedience with the license terms. As an example of this, the MIT and Apache licenses, demand that attribution is part of the redistribution of the software. This encourages trust and integrity in the open source community.

Contributor License Agreements (CLAs)

Contributor License Agreements are an important tool that play a crucial role in the open source development by stating the terms and conditions by which contributors submit their

work on the open source projects (Wikipedia, s.d.). They serve as key legal tools in the ever-changing dominion of open source development. Clarity and consistency are needed when different contributions to an open source software are managed. Thus, CLAs assistance in defining the legal terms and covering various legal aspects such as ownership or licensing compatibility is highly crucial. These agreements serve as a foundation of legal guidance and protection for the project from one side and the contributors from another.

The legal aspects covered by CLAs that aid to establishing clear procedures are ensuring that contributions are in accordance with the terms of the license governing the project, setting criteria for accepting contributions and outlining the ideals and rules that contributors must stick to when contributing to the project (Github, s.d.).

Further that the legal aspect, Contributor License Agreements affect how contributors are credited for their work by giving structure for the recognition that contributors receive for their work. This motivates contributors and gives them value as the CLAs show that the role of every contributor is important part of the whole open source project. This aspect improves the collaboration in the community, encourages teamwork and increases the incentives for innovation. In short, CLAs secure fair play in the open source community which aids into creating a community driven system base on transparency and team work. They address legal issues, licensing issues and make sure that every contributor get the appropriate credit for their work.

License Compliance, Compatibility and Conflicts

Following the guidance from licenses and respecting the terms established is paramount when it comes to contributing to open source software (Fossity Blog, s.d.). This involves respecting the specific procedures and standards that promote transparency, obeying the regulations and the collaborative culture in the communities. Every open source license has comprehensive code inspection, corresponding documentation as well as obedience to license notifications. Respecting these three components equalizes with avoiding any legal difficulties.

Compliance with a project`s license involves frequent and ongoing deep analysis of the codebase. Often time automated tools are used in order to do regular audits, which play a crucial role in identifying and fixing any non-compliance aspects. From the practical standpoint, history has shown that developers must be especially cautious when integrating third party components in their projects, as often times these components do not fit with the license terms of the original project, which then leads to legal trouble.

Mixing components with different licenses in one project is another issue that has been repeated through history (MergeBase, s.d.). This phenomenon takes place when the terms of one license are contradicting with the terms of another license. Only when the license terms are in sync – the licenses can be used on the same project. Contributors must ensure that the are using compatible licenses that work together without creating problems.

Understanding the compatibility of licenses and the potential conflict arising from their incompatibility is important for the operating of open source communities. Consequently, education is needed on how to read different licenses as some licenses naturally work well together, allowing various codes to coexist without creating issues. Legal analysis is needed in order to avoid combining code governed by incompatible licenses.

Legal Education and Community Communication

Educating the contributor about the legal side of open source if of essential importance for all involved sides. This means helping contributors understand the rules that guide the process and secure fair play. It doesn't only involve staying compliant, but also proactively managing any potential legal issues. Educating contributors on the often-tricky legal aspect of open source licenses represents a huge aid in the direction of giving contributors the necessary skills, and knowledge necessary to handle licensing, intellectual property and other related legal aspects.

The necessary education can be obtained through various workshops, webinars and extensive educational material about licensing, copyright and intellectual property in general. The existence of online forums where not only the existing information but the new modified or adjusted rules and regulations are regularly shared between all sides is of great importance. By spreading awareness on how to choose the right licenses and how to understand the various legal models, open source projects cam lower legal risks and set the path for future success.

Teamwork mindset is necessary not only when collaborating on open source projects, but also on the legal matters guiding these projects. Bringing the community together into legal discussions and decision making builds a sense of teamwork and shared responsibility by giving everyone a say in legal policies. It is crucial for the future that the legal policies align with the core values of the community and its future goals.

The above-mentioned tools and documentations that exist to bring the legal education closer to the contributors must be easily accessible to all. This includes various legal documents, guidelines as well as specific contact and places that give answers to specific legal questions. Overall, combining legal education and community efforts is a great way of securing the sustainability of the open source projects. Legal education aids contributors in making the right

decisions and avoiding costly mistakes, while involvement of the community in the legal processes ensures that knowledge is distributed correctly and then properly put to practice. This is the only recipe for success in the ever-evolving realm of open source software.

8. Enforcement of Open Source Software

The main aspect of intellectual property law that govern the correct enforcement of open source software are copyright and licensing. When sharing their code with the open source community, creators still own copyright and can grant licenses for use. These two IP rights give them the power to control how their work is used.

Copyright law protects the rights of contributors and keeps open source projects collaborative by preventing unauthorized sharing of the codes or changing and adjusting the projects (O'Reilly Media). On the other hand, open source licenses set out the rules for software use. Licenses ensure that the rights of the contributors are protected and while doing so – promote collaboration. They do so by guaranteeing transparency, dispute resolution, code sharing and monitoring compliance.

Open source software is based on a decentralized nature which can only be regulated if the collaborators proactively participate in the enforcement of the governing intellectual property rights (O'Reilly Media). Anyone who does not report non-compliance to their rules, shares the responsibility for the consequences taking place due to the illegal actions of others. Encouraging reporting of non-compliance is crucial.

Enforcing intellectual property rights in open source brings confidence to contributors that their work is used as intended, encouraging more results and more stability as users trust in a software project that offers reduced legal uncertainties and disruptions in the progress. Healthy open source community from a legal standpoint attracts more contributors and eventually faster advancement.

9. Conclusion

Exploring and studying open source software and intellectual property exposes a complex, dynamic and everchanging landscape. This is reflected through the connection between open source software and patents, along with licensing and intellectual property policies of companies. Legal battles between different entities highlight issues for the open source community and show us the importance of having a strong legal system that governs it. Looking into the future, developments in technology, cooperation and adjustable laws, if used correctly, will shape open source communities in the right way. The usage of blockchain and

advanced Digital Rights Management (DRM) is a big novelty in the balancing of innovation and protection. Looking forward, the future of open source and intellectual property rights mostly will depend on finding and keeping the perfect balance between new ideas, collaborative mindset and protecting creativity in the digital space. This can involve dealing with digital issues through tackling concerns about ownership, utilizing open source licenses and ensuring that proactive efforts from the communities are made.

In the following years, the relationship between open source software and intellectual property will evolve. Artificial intelligence, blockchain and augmented reality will bring to attention new questions about ownership and with that, new regulations that will guide these segments will be needed. When implementing these new technologies into open source, there will be a need for striking a balance between the promotion of new ideas and making sure that corresponding intellectual property rights are being respected. Digital material will have to be kept safe and technologies from the realm of digital rights management (DRM) will play a crucial role in this. With the increased usage of digital content, platforms should invest in better digital rights management in order to protect their property online.

The main role of blockchain technology when it comes to the relationship between open source and intellectual property will be to keep a record of ownership, meaning who own what and it will define the origin of the ownership (Merkle Science, s.d.). IP management will be made simple and arguments on the same topic will be avoided with the utilization of smart contracts. There will be a need for increased cross country collaboration on the topic of open source, specifically on adjusting and equalizing the IP protection rules. This can be done through agreements which will make sure that everyone follows the same rules, even across borders.

With technology evolving with a rapid pace, adjustment of laws must follow in parallel. The quality of the new laws will depend on the teamwork between the creators of the law and the open source contributors which would mean finding the right balance between these two sides in order to create laws that will work perfectly with the rapidly evolving technology. Legal cases such as The SCO Group vs. IBM, Oracle America, Inc. v. Google LLC, and Jacobsen v. Katzer are a great example of what might be ahead of us. While we learn from such cases, we must be prepared to face new issues that will be representing the new era of the relationship between open source and intellectual property. The only correct way of approaching the issues will be to look at the big picture, from sticking to the regulations that govern this phenomenon through employing all sides when a problem arises. With the rapid growth of open source projects, balancing creativity innovation and intellectual property rules through mutual collaboration will creative the perfect synchronicity in the ever-changing digital world.

Bibliography

European Union. (2023). *Contribution Agreement Manual.* pp. 4-30

Fossa. (2021, June 3). Retrieved from All About Permissive Licenses: https://fossa.com/blog/all-about-permissive-licenses/

Fossity Blog. (n.d.). Retrieved from Understanding Open Source Licenses Compliance: A Guide for Developers and Organizations.

Github. (n.d.). Retrieved from Contributor License Agreement (CLA): https://github.com/github/github-ospo/blob/main/policies/contributor-license-agreements.md

Gomulkiewicz, R. W. (2009). Open Source License Proliferation: Helpful Diversity or Hopeless Confusion? *Washington University Journal of Law & Policy*, pp. 279-283.

GOOGLE LLC v. ORACLE AMERICA, INC, No. 18-956 (Supreme Court of the United States April 5, 2021). pp 1-62

Heffan, I. V. (1997). Copyleft: Licensing Collaborative Works in the Digital Age. *Stanford Law Review*, pp. 1-6.

Jacobsen v. Katzer (United States District Court for the Northern District of California February 16, 2010).

Majerus, L. (2006). *Patent rights and open source—can they co-exist?* Fenwick & West. pp 1-3

MergeBase. (n.d.). Retrieved from Everything You Need to Know About Open Source License Compliance: https://mergebase.com/blog/open-source-license-compliance/

Merkle Science. (n.d.). Retrieved from How blockchain data can be leveraged by law enforcement agencies: https://www.merklescience.com/how-blockchain-data-can-be-leveraged-by-law-enforcement-agencies

O'Reilly Media. (n.d.). Open Source Licensing, Contract, and Copyright Law. pp 3

O'Reilly Media. (n.d.). Open Source Licensing, Contract, and Copyright Law. pp- 4-7

Schryen, G., & Kadura, R. (n.d.). *Open source vs. closed source software: towards measuring security.* Universität Paderborn.

SCO Group v. IBM, No. 16-4040 (Court of Appeals 10th Circuit 2017).

Shagall, Y. (2008). *Jacobsen v. Katzer: Federal Circuit Affirms Economic Interest of Open Source Copyright Holder.* Jolt Digest.

Smith, B. L., & Mann, S. O. (2004). Innovation and Intellectual Property Protection in the Software Industry: An Emerging Role for Patents? . *The University of Chicago Law Review*, pp. 262-264.

Smith, P. M. (2022). Copyright, Contract, and Licensing in Open Source. Oxford Academic. pp 71-15 84-87 87-90

Starita, A. (2023, November 30). *Mend.io*. Retrieved from Top Open Source Licenses Explained: https://www.mend.io/blog/top-open-source-licenses-explained/

The SCO Group, Inc. v. International Business Machines Corporation (United States Court of Appeals for the Tenth Circuit October 30, 2017).

Wikipedia. (n.d.). Retrieved from Open-source license: https://en.wikipedia.org/wiki/Open-source_license

Wikipedia. (n.d.). Retrieved from Contributor License Agreement: https://en.wikipedia.org/wiki/Contributor_License_Agreement

World Intellectual Property Office (WIPO). (2020). What is Intellectual Property?