DOI: 10.1111/jwip.12114

ORIGINAL ARTICLE

WILEY THE JOURNAL OF WORLD INTELLECTUAL PROPERTY

Free and Open Source Software and FRAND-based patent licenses

How to mediate between Standard Essential Patent and Free and Open Source Software

Catharina Maracke[1,2,3]

[1] Berkman Klein Center for Internet & Society, Cambridge, MA, USA

[2] Keio University, Shonan Fujisawa, Japan

[3] Kühne Logistics University, Hamburg, Germany

Correspondence
Catharina Maracke, Kühne Logistics University, Großer Grasbrook 17, 20457 Hamburg, Germany.
Email: cmaracke@law.harvard.edu

Funding information
Kühne Logistics University

In the context of digital transformation, both standards and Free and Open Source Software (FOSS) can be seen as key to success for innovation. However, the increasing role of FOSS in the ICT sector has provoked the question of interplay and compatibility between FOSS and standardization processes. The goal of this article is to provide an overview of the different licensing regimes, FOSS licenses and fair, reasonable, and non-discriminatory (FRAND)-based licenses, and to shed some light on the current debate by analyzing possible tensions between some of the most prominent FOSS licenses and their interaction with FRAND-based patent licenses. The article will begin with an overview of the history and context of FOSS licenses and FRAND-based licensing regimes, continue with a legal review of the actual language of prominent FOSS licenses, and provide further thoughts on future opportunities and challenges for the interaction between FOSS projects and standard processes, with a particular view on FOSS implementations of standards.

KEYWORDS
FRAND licenses, Open Source Software, Standard Essential Patents

1 | INTRODUCTION AND CONTEXT

While the Free and Open Source Software (FOSS)[1] development model has been tremendously successful in facilitating software innovation (Black Duck's & North Bridge's Survey, 2015), recent discussions in the context of standards have raised the question of whether Standard Essential Patents (SEPs) and especially the patent owner's interest in seeking royalties create a conflict with FOSS development.

Some advocates and scholars have argued that the FOSS licensing model is incompatible with the Fair, Reasonable and Non-Discriminatory (FRAND) licensing model facilitated by Standard-Development Organizations (SDOs). More precisely, many critics focus on the absence of licensing royalties as one of the key criteria for the FOSS definition(s) and thereby claim that standards including FOSS components should be royalty-free as well.

Part of the concern can be addressed by analyzing the legal details of the different licensing regimes. However, another part of the debate seems to be rooted in politics or culture. The narrative of how FRAND licensing models discriminate against FOSS licensing models is often supported by the general criticism of patent holdup or encouraged by the even more heated and emotional debate around the question of whether software should be patentable at all.

As a policy matter, it should be noted that both SDOs and the FOSS community have contributed enormously to the technological achievements of the past decades (Kappos, 2017, p. 259).[2] In fact, most commentators agree that FOSS projects need standards, and that standards need FOSS implementations, particularly in the context of the "Internet of Things" (Dapp & Bernauer, 2009). The European Commission has recently mentioned the importance of standards for innovation and growth and also emphasized FOSS development as another driving factor for innovation (European Commission, 2017). Accordingly, the Commission asked for "flexible and effective interactions between standardization and open source communities" assuming that such flexible and effective interactions will "promote and accelerate the uptake of advanced technology development" (European Commission, 2017, p. 12) (Dapp & Bernauer, 2009, pp. 1–26). Relatedly, in a recent Communication entitled "Setting out the EU approach to Standard Essential Patents," the European Commission has not only outlined the importance of standards for innovation and growths and called for a clear, balanced, and reasonable policy for SEPs in Europe, but has also acknowledged FOSS development as another driving factor for innovation (European Commission, 2017). Consequently, the Commission emphasizes "flexible and effective interactions between standardization and open source communities" assuming that such flexible and effective interactions will "promote and accelerate the uptake of advanced technology development" (European Commission, 2017, p. 12).[3]

How can such flexible and effective interactions be initiated and facilitated? The FOSS community has developed a unique and well-functioning governance system to develop and maintain FOSS and facilitate related licensing questions. Is it possible to pave the way for effective interoperability and practical and cultural integration between traditional patent innovation and open innovation approaches?

The goal of this article is to provide an overview of the actual debate between current practices in the standards ecosystem and corresponding arguments in the open source community. While there are different ways of interaction between standards and FOSS, including the integration of FOSS in the development of a standard or the more general concept of Open Standards,[4] the article will focus on the question whether and under what conditions a standard can be implemented as FOSS. Beginning with the historical context and legal analysis, the following paragraphs will investigate new paradigms currently discussed by SDOs and its industry members and explore concerns raised by the FOSS community.

2 | FREE AND OPEN SOURCE SOFTWARE

2.1 | History and context

Most of the software created in the 1950s and 1960s IBM (1960) was "open" in the sense that it was developed by computer science academics and corporate researchers working together and distributing results under principles of

transparency, sharing and cooperation (Perens, 1999 p. 1). Computer hardware was built as large and expensive machines, operated in air-conditioned computer rooms, and accompanied by additional services and software without additional or separate charges.

With the development and spread of personal computers in the early 1970s, a new business model emerged based on the understanding that software is covered by copyright, which along with contract law provides a legal basis to establish exclusive rights.[5] Rather than offer hardware, services and software exclusively in packages, marketers "unbundled" hardware and software components and offered them separately and under different conditions. To complete the business model of using copyright in software as a property to be licensed independently from hardware components, the distinction between the human-readable source code and the machine-readable object code became a critical aspect for most distributors. Even today, most commercially licensed software is distributed in machine-readable object code, whereas the human-readable source code is kept proprietary and not disclosed in order to prevent competitors from identifying and copying the methods embodied in the code.[6]

The fact that proprietary software products are not only subject to payment models but also closed to the user to understand and modify has opened a heated debate, which includes a variety of perspectives and positions, whereby supporters of FOSS focus on promoting software as "free" in the sense that it is publicly available with fewer restraints as to source code access and further distribution, but not necessarily in the sense that it is "costless."[7]

"Free" software was the term promoted by Richard Stallman, a member of the Artificial Intelligence Lab at MIT in the 1970s and early 1980s (Fogel, 2016). Stallman was especially concerned that he could no longer study and modify software programs that were originally written by others. In his view, the practice of distributing only the executable machine code that was compiled from the source code was ethically wrong. In response, he decided to resist and started the Free Software Foundation as an organization to promote the notion of free software.[8] Stallman's vision was to develop a completely free and open computer operating system and body of application software, in which users and developers would be completely free to share, modify and further distribute the code.[9]

Besides the new operating system, Stallman also worked with a group of legal experts to create a new concept of copyright licenses to be used for free software. Instead of the traditional licensing model, which usually requires the payment of royalties in exchange for exploitation of the respective work, he devised a license with a twofold purpose. First, the terms of the new license would ensure that the code is available to be studied, modified, and shared; and second, these terms would be subject to a few conditions, most importantly, if the licensee distributed any of the code in any form, she would be required to make copies and modifications of the source code available under the same license as the original and without any further restrictions (Fogel, 2016). In effect, Stallman and his legal team used copyright itself as a legal mechanism to guarantee that all code licensed under this license would be perpetually available to be studied, modified and further distributed. For Stallman, this new concept was more important and more effective than simply allowing his code to be shared and modified without the additional legal conditions, because it would prevent the possibility of incorporating the code into a proprietary program.

The legal trick that created a form a protectionism for free software, better known as copyleft, turned out to be one of the most important but also most controversial elements for all future generations of free software licenses. Today, copyleft remains an essential criterion for the GNU Public License (GPL) family and few other licenses, such as the Eclipse Public License 1.0 (Reference: https://www.eclipse.org/legal/epl-v10.html) or the European Union Public License (Reference: https://opensource.org/licenses/EUPL-1.1). However, a variety of other FOSS licenses, such as the Berkeley Software Distribution (BSD) license, the MIT license, or the Apache license, allow greater flexibility with regard to the modification and sharing requirement.

2.2 | Definitions: Free Software Foundation and Open Source Initiative

Based on Stallman's vision that software should be free, the Free Software Foundation defines free software as software that gives a user the freedom to (i) run the program for any purpose; (ii) study how the program works and

change it; (iii) redistribute copies and (iv) distribute copies of modified versions to others.[10] While the concept of copyleft is an important feature to protect these four freedoms, it is not a necessary requirement to qualify software projects as free software. As a matter of fact, the Free Software Foundation explicitly states that there are important reasons why it is better to use copyleft,[11] but recognizes non-copylefted free software as being ethical, too, according to the official free software definition (Free Software Foundation, 2018).[12]

While the general idea of free software became successful, the terminology and especially the social activism to promulgate the concept did not appeal to the commercial world.[13] One challenge was the terminology. Because the English language does not differentiate between low price and liberty (or gratis and libre in other languages), the term free software invariably required further explanation.[14] Another problem was the tone of social and political activism. The term "free" was promoted in a way that saw freedom as an end in itself, it did not matter whether free software also happened to be better in terms of quality, or more profitable for certain businesses under certain circumstances (Fogel, 2016).[15]

As a result of the debate surrounding terminology, interactions with the corporate world, and activism, a group of programmers in the free software movement created the term "open source software" as an alternative label (Jaeger & Metzger, 2016, p. 9).[16] Using the term "open source" rather than "free" was not meant as competing philosophies but rather as a different method to market the same concept (Perens, 1999). Hence, introducing a new term was primarily a strategic decision, a way to "promote the idea of free software to business people".[17] Like the Free Software Foundation, the Open Source Initiative[18] established a set of criteria for licensing terms to qualify as open source conformant, in short: i) free redistribution of the software; ii) in source code and complied form; iii) allowing modifications and derived works, including the possibility to be redistributed under the same terms as the license of the original software; iv) integrity of the author's source code; v) no discrimination against any groups or persons and vi) any fields of endeavor; vii) further distribution without the need for execution of an additional license; viii) with the license not being specific to a product and ix) not restricting other software and x) being technology–neutral, meaning not predicating on any individual technology or style of interface.[19]

Since the Free Software Foundation strongly objected to the open source approach and terminology, the free and open source ecosystem is being decided in its terminology. However, the ultimate goal of both the free software movement and the open source software approach is to make the software available under legal terms that allow and facilitate studying, modifying and further distributing the software.[20] Against this background, both definitions of "free" and "open source" software require that the respective source code must be available and that modification must be allowed. However, the detailed terms of each FOSS license can differ tremendously in the extent to which they impose additional requirements on the licensee.

2.3 | License taxonomy

A detailed look into the terms of FOSS licenses reveals that they can be divided into three categories: (i) strong copyleft licenses; (ii) weak copyleft licenses and (iii) permissive licenses.

Strong copyleft licenses require further licensees, who modify the software, to make such modifications available under the same terms under the original software was released without any further restrictions (Jaeger & Metzger, 2016, p. 41).[21] Weak copyleft licenses include additional preconditions to inherit the copyleft effect, for example, only if the original code is modified and distributed within the same file, does the copyleft effect have to be applied, whereas if any modification to the original code has been made in a separate file, then such separate file may be distributed under a different license (Jaeger & Metzger, 2016, p. 95 et seq.).[22] Permissive licenses give the user the greatest flexibility in the sense that they do not restrict the use of the software and do not impose any conditions as to which the software itself or any modification have to be used and subsequently licensed. Hence, software licensed under permissive FOSS licenses may be modified and distributed under any other license, including non-FOSS licenses, for example, proprietary licenses (Jaeger & Metzger, 2016, p. 102 et. seq.).

Within this broad categorization of FOSS licenses, there is another distinction to be made and that is with regard to the patent grant. Some FOSS licenses incorporate explicit patent grants, whereas others do not mention patents at all. Even within the category of FOSS licenses that include a patent grant, there are different nuances in terms of scope and structure. Some FOSS licenses have coupled the patent grant with a termination clause that immediately terminates the license in the event that the licensee institutes proceedings for patent infringement.[23] Others simply comprise a patent license without termination clause.[24]

In particular, FOSS licenses without an explicit patent grant have encouraged a discussion whether those licenses include an implied patent license based on the argument that only an implied patent license would give the licensee the necessary rights to make use of the copyright license. In other words, if there were no implied patent license included in the FOSS licenses, so the argument supporting this view contends, then the copyright grant to use and redistribute the software would not make sense seeing as the potential patent owner could always prohibit the use of the software licensed under a copyright license. Because of the way software interacts not only with copyright but also with potential patents, a patent license is needed to facilitate the use enabled by the copyright license. And whenever such a patent license is not explicitly mentioned, supporters of this argument read an implied patent license into the FOSS license (Jaeger & Metzger, 2016, p. 104).[25]

However, the opposing view is that FOSS distributed and used under a FOSS license without an explicit patent grant is not protected by a patent license. The failure to include an express patent license might also mean that no patent license rights are granted (Kappos & Harrington, 2018).[26] Historically, some of the most common FOSS licenses, such as the GPL or the BSD, were written at a time when it was not at all clear that software was patentable.[27] Hence, these licenses focused on copyright to describe the rights granted and obligations imposed, but they did not focus on or include an express patent grant. A more detailed analysis of the exact language used in each license might support the view that patents should only be licensed when explicitly mentioned. The GPL 2.0, for instance, grants the right to

> *... copy and distribute (and modify) the code provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty*;

While patents are not included, they are also not expressly excluded. Nevertheless, where the GPL 2.0 addresses the question of patents, it does so expressly, for example, in the context of third party patents dealt with in section 7 of the license.[28]

Similarly, the BSD license grant simply allows

> *redistribution and use in source and binary forms, with or without modifications,*

provided that certain conditions are met, such as the retention of the copyright notice. Compared to the MIT license, which also uses patent terms such as “sell” and “deal in” to describe the rights granted in the license, there could be arguments to support the view that the MIT license is more likely to be deemed to also include patent rights, whereas the BSD does not include any patents (Nadan, 2009, p. 2).

Independent of this discussion, almost all scholars agree that the patent license, whether explicitly mentioned or implied through omission, relates only to the patent(s) relevant to use the respective software and only to the extent needed to use the software according to the accompanying copyright license.

Taking the language of the Apache 2.0 license as a prominent example of an explicit patent grant, section 3 constitutes a patent license by the granter to the licensee of any patents to which the granter has the right to license, coupled with a termination clause that immediately terminates the Apache's patent license grant in the event that the licensee institutes proceedings for patent infringement. However, the license grant is explicitly limited to those patent claims that are (i) licensable by the licensor (i.e. the Copyright holder) and (ii) necessarily infringed by the contribution:

> *Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.*[29]

A very similar, explicitly limited patent grant can be found in section 13 of the Perl Artistic License 2.0, which includes a license by the granter to the licensee of any patents to which the granter has the right, and which will be terminated automatically in the event that the licensee initiates an action for patent infringement:

> *This license includes the non-exclusive, worldwide, free-of-charge patent license to make, use, offer to sell, sell, import and otherwise transfer the Package with respect to any patent claims licensable by the Copyright Holder that are necessarily infringed by the Package. If you institute patent litigation (including a cross-claim or counterclaim) against any party alleging that the Package constitutes direct or contributory patent infringement, then this Artistic License to you shall terminate on the date that such litigation is filed.*[30]

Section 2 of the European Public License 1.2 sets this out even more succinctly:

> *The Licensor grants to the Licensee royalty-free, non-exclusive usage rights to any patents held by the Licensor, to the extent necessary to make use of the rights granted on the Work under this License.*[31]

Despite differences in terminology and potential termination, all reviewed patent licenses explicitly included in some of the FOSS licenses cannot extend beyond patent claims that are (i) held or licensable by the licensor and (ii) necessarily infringed by the package (Mitchell & Mason, 2011, p. 3). Consequently, it seems very likely that also implied patent licenses, if applicable, are restricted by these conditions.

3 | STANDARD ESSENTIAL PATENTS AND FRAND-BASED LICENSES

Standards are all around us and play a central role in virtually all economic and social activity (Blind & Kahin, 2018, p. 1). A standard can be defined as a set of rules or guidelines that guarantee interoperability. More precisely, standards set out the requirements for a specific item, material, component, system or service, or describe in detail a particular method or procedure.[32] Especially in the information and communication technology (ICT) industry, and in particular the Internet of Things (IoT) sector, where the economic potential of new products and applications is expected to grow tremendously over the coming years (European Commission, 2017),[33] it is critical to establish an interoperable technical framework that enables the implementation and combination of new inventions.[34]

While standards can be developed in a formal way, typically facilitated by Standard Development Organizations (SDOs) in a complex institutional and technological process, discussed and agreed upon by their different member organizations (Gupta, 2018, pp. 31–42),[35] they can also be established by way of widespread use and broad market shares of a new technology, becoming so called *de facto* standards (Ménière, 2015, p. 7).[36]

3.1 | Standard essential patents

Since standards, formal or de facto, facilitate the deployment of new technologies, they are subject to legal rights, and especially intellectual property rights, which act upon not only their development but also their implementation (Fitzgerald & Pappalardo, 2009, pp. 467–483). A single standard in the ICT sector may be covered by hundreds of intellectual property rights, especially patents. These patents are "essential to the standard" because anyone implementing the standard will necessarily use that patented technological feature in order to make, use, or sell a product or provide a service which complies with that standard (Nilsson, 2017, pp. 1017 and 1019). Consequently, any implementation of the standard requires a patent license from any contributor to the relevant standard who owns an essential patent in the standard.

Against this background, the technological complexity of today's standards in the ICT sector can easily cause an imbalance and tension between the interest of the patent owner and the interest of the implementer. Standards, on the one hand, are designed to be shared. The objective of standards is widespread adoption of the underlying technologies. Hence, their benefit is directly related to the number of users and the frequency of use (Blind & Kahin, 2018, p. 1). Patents, on the other hand, according to the most conventional patent theory, are designed as exclusive rights to reward inventors with an exclusive market position for a number of years (Haedicke & Timmann, 2014, p. 6). As a result, SEPs give rise to potential friction between individual property rights and the rightholders' interest to exploit their full potential and the public interest to share standards and make them available to be used and implemented with great frequency. In order to address this conflict and to avoid a situation where the SEP holder blocks the standard by refusing to license or demanding excessive royalties and at the same time guarantee that the SEP holder obtains a timely return on investment, nearly all SDOs have developed and defined Intellectual Property Rights (IPR) policies by *consensus* between their members (Ménière, 2015, p. 8). The core element of almost all SDO's IPR policies is that they require their members to offer access to the respective SEP under conditions that are "fair, reasonable, and non-discriminatory," that is, "FRAND" (Nilsson, 2017, p. 1019). FRAND commitments deliver on both aspects: Implementers of the respective standard can get licenses at fair and reasonable terms to allow for widespread adoption and SEP holders can receive fair and reasonable remuneration for their patented inventions.

As much as the FRAND licensing policy has helped consumers to enjoy interoperable and innovative products, the flexibility to determine the FRAND terms and conditions has also led to significant legal uncertainty as to the exact scope and requirements of FRAND-based licenses.

3.2 | FRAND licensing terms

While initial approaches to the FRAND principles can already be found in mid-twentieth-century U.S. Antitrust Decrees, the American Standards Association (ASA) can be considered to have published the first FRAND licensing policy in 1959.[37] According to the ASA policy, "standards should not include items whose production is covered by patents unless the patent holder agrees to and makes available to any interested and qualified party a license on reasonable terms" (ASA 1959 policy, 11.6). A less detailed policy with regard to today's FRAND principles was provided by the American National Standards Institute (ANSI) in 1970, which stated that "a license will be made available to applicants under reasonable terms and conditions" (ANSI 1970 policy, 86.1).[38]

Similarly, and beyond the U.S. context, an early approach by the International Standards Institute (ISO), the International Electrotechnical Commission (IEC), and the early European organizations such as the European Committee for Standardization (CEN) and the European Committee for Electrotechnical Standardization (CENELEC) was to allow patents in standards as long as the patent holders were "willing to negotiate licenses . . .. with applicants throughout the world on reasonable terms and conditions" (Contrereas, 2018, p. 163). More precisely, the European Telecommunications Standards Institute (ETSI),[39] like most other SDOs policies available today,[40] arranged that the

"availability of licenses" would be on "fair, reasonable and non-discriminatory (FRAND) terms and conditions" (ETSI Intellectual Property Rights Policy, 6.1).

With the growing number of SEP's and SEP holders during the 1990s and 2000s, the question of IPR policies became even more complex. More SEP holders negotiating with more and more implementers resulted in a growing number of licensing agreements per standard (Ménière, 2015, p. 9).

Since the Court of Justice of the European Union[41] has set a framework for the process of SEP negotiations between the parties, the wide-ranging discussion about FRAND licensing focused especially on the question of what exactly FRAND means in each individual case.[42] The general meaning of the single elements that form the acronym FRAND seem to be quite clear with "fair" in the sense that equal considerations on both sides have been considered, "reasonable" in the sense that both parties consider the outcome of the negotiations as acceptable on business terms, for example, the payable rate reflects the value of the license, and "non-discriminatory" in the sense that neither side is treated detrimentally compared to any other similarly situated party.[43] Nevertheless, within these parameters, the final decision about what constitutes as fair, reasonable and non-discriminatory in the patent licensing context is left to the SEP holder and the implementer to be determined in each individual case and their bilateral license negotiations.

4 | THE INTERSECTION OF STANDARD ESSENTIAL PATENTS AND FREE AND OPEN SOURCE SOFTWARE

The question of software patents has been an area of considerable contention, particularly in relation to whether patents can or should be able to coexist with FOSS licenses (Mitchell & Mason, 2011, p. 1).[44] The issue has received a great deal of attention with regard to the intersection of SEP and FOSS and especially the question of compatibility between FOSS licensing terms and any standards containing SEPs. While there are different scenarios of SEPs interacting with FOSS, such as the possibility for SDOs to host FOSS projects within a standardization process, whereby a new and open standard will be created,[45] or the initial implementation of software via FOSS projects, which eventually leads into technical specifications of a standard,[46] the most critical questions arise in case of a FOSS implementation based on (existing) standards, whereby the standard includes SEPs and the patent holder is a third party. Whenever a software developer writes software to operate with a standard including an SEP, that development can lead to an infringement of the patent, unless the patent holder gives permission and grants a patent license to the developer. Assuming the patent holder has committed to license the patent under FRAND conditions, these conditions may interfere with the conditions of the desired FOSS license covering the software to be developed.

Looking at today's innovation landscape, the transformation of traditional industries into technology companies, and the success of software companies,[47] an increasing convergence of patents and software can be predicted, whereby many programs and projects will include FOSS. A 2015 survey shows that 78% of all companies run part or all of their operations on FOSS (Black Duck's & North Bridge's Survey, 2015).[48] Hence, the question of SEP interacting with FOSS and the question of compatibility between the two licensing models is not only a theoretical issue but a practical one as well.

4.1 | Compatibility of FRAND-based licenses and FOSS licenses

Some commentators have claimed that FOSS developers are unable to implement FRAND-based standards and distribute impacted products because FRAND terms conflict with FOSS license obligations.[49] The essence of the conflict as defined by those commentators is that a FOSS licensee has the right to distribute the product without having to pay royalties, which conflicts with the requirement of the interfering FRAND license to pay some kind of (FRAND) royalty.[50] Consequently, according to this view, the only way to overcome the dilemma would for be for SEPs to be royalty free.

Going back to the determination of FRAND terms and especially the question of licensing rates, it seems quite obvious that there has to be at least some sort of range between a minimum and a maximum fee to be negotiated between licensor and licensee in order to qualify as fair, reasonable and non-discriminatory.[51] However, it cannot be considered unreasonable *a priori* if the patent holder decides to award a royalty-free license and not to charge any licensing rate at all. Argumentum a fortiori it would be apparently reasonable in the given sense if licensor and licensee both accept a royalty-free license. The fact that FRAND is a vague concept leaves room for the parties to negotiate, whereby the parties have the flexibility to mutually agree on royalty-free access to the respective technology. Such a royalty-free license can also be defined as non-discriminatory if neither side is treated in detriment compared to another party similarly situated (Nilsson, 2017, p. 1017 et seq.). Following the above mentioned general meaning of FRAND, this does not necessarily mean that the patent holder has to award a royalty-free license to everyone after he granted such a license to one specific licensee. Only in case the implementers of the standard are in similar situations, they must be able to obtain a royalty-free license as well. In summary, the legal interpretation of FRAND licensing terms cannot exclude *a priori* the possibility of a zero-licensing rate.

Nevertheless, it appears questionable whether a zero-licensing rate is the only way to enable compatibility of FOSS licensing terms with FRAND licensing terms, and—more importantly—whether royalties are the only obstacle to compatibility between SEP and FOSS licensing models.

The requirement of FOSS licenses to make the source code available on a royalty-free basis only means that the licensor may not require the recipients of the code to pay the code distributor any royalties. It does not prevent the developer or code distributor from taking a patent license from any third party and paying per-unit royalties on the copies of the code it distributes under a FOSS license, as implementing a FRAND based standard might require (Kesan, 2011, p. 4). That being said, the requirement to pay (any kind of) royalties to the patent owner does not conflict with the license grant of FOSS licenses to offer the source code on a royalty-free basis for use, modification and re-distribution. The critical issue of interoperability is the structure of the patent grant itself. The patent owner will license the patent to an individual licensee on individual terms (on FRAND terms, which can include zero royalties or any other individually negotiated rate), but the licensee will not be allowed to sublicense or assign the benefits of the license to any other party (Mitchell & Mason, 2011, p. 3). Consequently, any downstream user of the software interacting with the respective patent would have to obtain a separate and individually negotiated patent license from the patent owner.

Even if this formation were theoretically possible and could potentially be conformant with the general meaning of FRAND licensing terms as indicated above, there may still be inconsistencies with the fundamental architecture of at least some of the most popular FOSS licenses.

4.1.1 | Strong copyleft licenses

Strong copyleft licenses require the licensee to preserve the same rights in any derivative work.[52] Taking the GPL 2.0 as an example of the most popular copyleft licenses,[53] section 2b states:

> *You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.*[54]

As indicated above, there might be reasonable grounds to limit the license grant to copyright.[55] Following this view and reading the wording in isolation from the remaining text of the GPL 2.0, it might be argued that this copyleft section refers to the copyright license, meaning that the work should be (copyright) licensed as a whole to all parties and without any charge. However, it seems difficult to ignore other parts of the license while interpreting the copyleft clause and especially difficult to ignore section 7 which states:

> *If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.*[56]

Notably the last sentence explicitly mentions patents and prevents any redistribution that is not royalty-free and that cannot assure further royalty-free redistribution by anyone who receives a copy of the respective program. Even if FRAND terms can be negotiated to include zero royalties,[57] there may be another issue related to the structure of the patent license: Given the context and already existing tension between patents and standards in general,[58] the FRAND license will most likely be limited to the purpose of implementing the standard. Hence, the licensee under the GPL 2.0 license may be unable to license the work "as a whole at no charge to all third parties," in particular to those third parties who might seek to develop the software for a reason other than interacting with the standard.[59]

Even in the unlikely event that the FRAND license itself is drafted to be royalty-free and includes a very broad patent grant allowing the use of the patents to all parties and for all purposes, the language of section 2b will most likely create a conflict with FRAND licensing terms: The structure of section 2b sets out three requirements, which are not alternative but cumulative. First, the work as a whole should be licensed; second, the license of the work as a whole should be without royalties and third, the work should be licensed to all parties. If it can be assumed that the work as a whole includes that part of the work, which would in absence of a patent license infringe the respective patent, then the license of the work as a whole should include a patent license in respect of that third-party patent (Mitchell & Mason, 2011, p. 6). This, however, will create a legal conflict because the FOSS licensor will not be able to license third-party patents. Even if the general view on potentially implied patent licenses is that those licenses are limited to patents owned or otherwise licensable by the (FOSS) licensor (Jaeger & Metzger, 2016, p. 58),[60] the wording of the copyleft section included in the GPL 2.0 creates a conflict if it requires the work to be licensed as a whole.

Since the GPL 3.0 contains a very similar copyleft clause, which can be found in section 5, the same arguments and comments evaluated in the context of GPL 2.0 apply for the GPL 3.0: Section 5 of the GPL 3.0 provides that:

> *You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:*
>
> *a)–b)*
>
> *c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy.*

Similar to the wording of the GPL 2.0, the GPL 3.0 also requires the work to be licensed (i) as a whole; (ii) under this license and (iii) to anyone who comes into possession of a copy. Hence, a substantial problem arises from the fundamentally different architecture of downstream licensing required by the GPL 3.0 and the individual direct licenses based on FRAND terms.

The analysis of the GPL 2.0 and 3.0 as one of the most prominent copyleft licenses might suggest that the copyleft effect *per se* creates an issue of incompatibility with FRAND licenses. But a closer look into the language of another

relevant copyleft license, the European Union Public License EUPL 1.1, reveals that the wording of the EUPL 1.1 copyleft section appears to be different from the copyleft included in the GPL 2.0 and 3.0:

> *Copyleft clause: If the Licensee distributes and/or communicates copies of the Original Works or Derivative Works based upon the Original Work, this Distribution and/or Communication will be done under the terms of this Licence or of a later version of this Licence unless the Original Work is expressly distributed only under this version of the Licence. The Licensee (becoming Licensor) cannot offer or impose any additional terms or conditions on the Work or Derivative Work that alter or restrict the terms of the Licence.*

Looking into the "terms of this Licence," section 2 of the EUPL 1.1 expressly states that

> *the Licensor grants to the Licensee royalty-free, non-exclusive usage rights to any patents held by the Licensor, to the extent necessary to make use of the rights granted on the Work under this License.*

Hence, the scope of the EUPL 1.1 is expressly limited to patents that are owned by the licensor. As a result, any third-party patents, licensed under FRAND or other terms to the licensor are excluded, which does not only prevent any legal conflict, but also allows the licensee to obtain his own license from the respective patent owner—be it on FRAND terms or otherwise (Mitchell & Mason, 2011, p. 10).

Since the revised version 1.2 of the EUPL mainly addresses compatibility questions with regard to the GPL 3.0 and future versions of the EUPL but still includes the same language with regard to the copyleft section, no legal conflict between FRAND licensing models and the terms of the EUPL 1.1 and 1.2 can be found.

4.1.2 | Weak copyleft licenses

Weak copyleft licenses refer to licenses where not all derived works inherit the copyleft effect. Historically, weak copyleft licenses were drafted and used for software libraries to allow other software to link to the library with the possibility for that (linking) software to be redistributed without any restrictions or obligations. Only changes to originally licensed software itself will become subject to the copyleft effect, whereas changes to the software that links to the library can be licensed under any other license without any restrictions. The goal was to allow software of any license to be compiled and linked against libraries licensed under a weak copyleft license without any restrictions or license obligations imposed.

Taking the LGPL 2.1 as an example, this license applies to the library and includes a copyleft section that requires for any modifications

> *the whole of the work to be licensed at no charge to all third parties under the terms of this license (section 2c).*

However, this copyleft requirement only applies to "works based on the library." According to section 5 there is an exemption for any program that

> *contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it.*

Such a program is called a "work that uses the Library" and is characterized as a work that

> *in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.*

In summary, the copyleft clause will only apply to works based on the library and not to any work that uses the library. Potential conflicts with FRAND licenses as outlined above for the GPL 2.0, will therefore only arise if the original software is modified. Changes to the software that links to library are exempt and will not create any issues or conflicts with regard to FRAND licensing terms.

Another prominent example to be mentioned in the context of weak copyleft licenses is the Mozilla Public License MPL 2.0. Unlike the LGPL licenses, the MPL was not designed for software libraries but for Mozilla software such as Mozilla Thunderbird and especially Mozilla Firefox, which is based on the Netscape Navigator (Jaeger & Metzger, 2016, p. 86). Section 3.1 of the MPL 2.0 includes a copyleft clause that states:

> *All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License.*

Similarly to the terms of the EUPL 1.1 outlined above, the terms of the MPL 2.0 outlined in section 2.1 grant the licensee a

> *world-wide, royalty-free, non-exclusive license under intellectual property rights (other than patent or trademark) Licensable by such Contributor [Licensor] to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit" including a patent grant referring to "Patent Claims of such Contributor [Licensor] to make, use, sell, offer for sale, have made, import, and otherwise transfer . . .*

Hence, the wording of the patent grant is expressly limited to patents licensable by the licensor and does not include any third-party patents.

In addition, the patent grant is further limited by section 2.3 and excludes

> *any code that a Contributor [Licensor] has removed from Covered Software; or for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or under Patent Claims infringed by Covered Software in the absence of its Contributions.*

This provision prevents those patents from being licensed under the terms of the MPL 2.0, which will be infringed by future third party modifications of the licensed software.

Given the expressly mentioned limitations in the patent grant included in the MPL 2.0, it is very unlikely that any conflict between FRAND licensing terms and the terms of MPL 2.0 will be identified even when the copyleft effect has to be applied.

4.1.3 | Permissive licenses

Permissive licenses, as outlined above, ensure that source code and corresponding object code may be used by the licensee with minimal restrictions. Consequently, their licensing terms only serve to ensure that source code and the corresponding object code may be used with minimal restrictions.[61] Usually, these restrictions mean that all copyright and license notices should be included in all copies of the software and that the software is provided without any warranty as to fitness for purpose and limitation of liability. Taking the MIT license as an example, the only requirement in addition to the usual disclaimer that the software is provided as is and without any warranty, is the following:

> *The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.*[62]

Hence, there is no conflict with regard to any potential patent licensing terms, whether they are based on FRAND conditions or other provisions. Even supposing an implied patent license is included in the MIT license or other permissive licenses, that patent grant would be limited to patents licensable by the copyright licensor and to the extent necessary to make use of the rights granted under the copyright license.

4.1.4 | Summary

When examining the legal language of different FOSS licenses, no general conflict between FRAND licenses and FOSS licenses can be found. There is not even a general statement to be made that FRAND licenses conflict with all FOSS copyleft licenses or all FOSS licenses including an explicit or implied patent grant. Thus, the question of whether a legal conflict between FRAND licenses and FOSS licenses may hinder the implementation of a standard, has be evaluated on an individual basis and depends on the concrete language used in the respective FOSS license.

There are strong arguments to support the view that a conflict may arise with regard to the GPL 2.0 and 3.0 as well as to the LGPL 2.1 and 3.0 in case their copyleft effect is inherited. Yet, the legal conflict is independent of the general understanding of the copyleft mechanism but related to the specific language of the GPL license family.

4.2 | New paradigms

In addition to the long-established and widely used concept of FOSS licenses, new paradigms have been established, including new licenses that have been drafted to accommodate specific needs and requirements.

4.2.1 | OpenAirInterface

OpenAirInterface in its original setting is a FOSS implementation of the 4G network system, created by a group of researchers within EURECOM (OpenAirInterface, 2018).[63] In 2014, the OpenAirInterface Software Alliance was set up as an "endowment fund" aimed at leading an open and global software alliance (OSA—OpenAirInterface Software Alliance).[64] While the software was originally licensed under GPL (version 2.0 and 3.0) OSA has relicensed the code base under a new license that was specifically drafted for this project. According to the OSA, the OpenAirInterface Public License (OAI Public License) 1.1[65] is

> *a modified version of Apache V2.0 License, with the modified patent clause that allows contributing parties to make patent license available to third parties under fair, reasonable and non-discriminatory (FRAND) terms for commercial exploitation. The availability of OAI code is free for non-commercial/academic research purposes. The main reason for this modification is to allow companies/individuals which own significant portfolio of patents to be able to contribute to the OAI source code and still be able to keep their patent rights. Such a license will allow close collaboration with 3GPP member companies while at the same allowing commercial exploitation of the code. We have tried to follow 3GPP FRAND policy on licensing in this regard. We expect that with this modification to the OAI Public License V1.1, it will allow more companies/individuals to contribute to the software, while protecting their essential IPR.*[66]

Following the terminology of the Apache 2.0 license, the OAI Public License1.1 includes a patent grant, with the details of the grant being modified depending on the purpose of the licensee.

In case the patent license is granted for study, testing and research purposes, the

> *Licensor and each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, and*

> *otherwise transfer (excluding selling) the Work, solely for study, testing and research purposes, where such license applies only to those patent claims licensable by Licensor or such Contributor that are necessarily infringed respectively by the Work and/or the said Contributor Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted ('Essential Patents').*

For any other purposes,

> *You commit to be prepared to negotiate a non-exclusive, non-transferable, non-assignable license of Essential Patents with each Contributor and/or the Licensor on Fair, Reasonable and Non-Discriminatory ('FRAND') terms and conditions for the use of the Work or Contribution(s) incorporated within the Work.*

In addition, the OAI Public License 1.1 also includes a termination clause stating that

> *If You institute patent litigation against any entity making use of the Work solely for study, testing and research purposes (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then the patent licenses granted to You under section 3.1 of this License for that Work shall terminate as of the date such litigation is filed.*

Clearly, the OAI Public License was designed to ensure full compatibility with any possible FRAND licenses. The fact that the patent grant is differentiated by the purpose of the license and explicitly includes a commitment of the licensee to negotiate a FRAND license related to any patents used for other purposes than study, testing and research, allows for compatibility with any possible FRAND licensing terms.

4.2.2 | Broadband Forum

The Broadband Forum, a non-profit industry consortium dedicated to developing broadband network specifications[67] has taken a similar approach. Section 3.1 of the Broadband Forum IPR policy requires all submitters to "provide a License to all Necessary Claims Owned by it or any of its Related Parties and included in its Submission that become Necessary Claim(s), at its option either with or without compensation and also on a RAND basis, to all Implementers."

A license is defined as

> *either (a) an agreement to license Necessary Claim(s) to any Implementer, on a perpetual, non-exclusive and worldwide RAND basis (either with or without compensation), permitting the licensee to make, have made, use, reproduce, market, import, offer to sell and sell, and to otherwise distribute products that implement such Necessary Claim(s), or (b) a binding, perpetual, irrevocable commitment not to assert Necessary Claim(s) against any such Member or non-Member Implementer.*[68]

In addition, any software must either be submitted by way of transferring copyright ownership to the forum or by granting the forum and any implementer a BSD-3 clause license.[69] However, the Broadband Forum requires that patent claims contained in the respective software that are necessarily infringed by implementation

> *shall be subject to the patent licensing requirements of section 3.1. Further clarification is provided by stating that no other express or implied licenses to any party's patent rights are granted by this section 4.3.*[70]

In essence, the forum's IPR policy demonstrates compatibility between FRAND licenses and the permissive FOSS licenses par excellence. Assuming that the BSD does not include an implied patent license, any patent necessarily infringed while contributing and modifying the software will be subject to a FRAND license or a non-assertion pledge with or without compensation.[71]

4.2.3 | ETSI Open Source MANO

Open Source MANO is an initiative to develop an Open Source NFV Management and Orchestration (OSM) software stack hosted by ETSI and aligned with ETSI NFV.[72] The OSM terms of reference state that all code will be made available under the Apache 2.0 license. In addition, all contributions shall be made under the Apache 2.0 license and the respective corporate or individual contributor license agreement.[73] The contributor license agreement reflects the language of the copyright and patent license included in the Apache 2.0 licenses and therefore includes a

> *perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to the project manager and to recipients of software distributed by the project manager to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute your contributions and such derivative works.*[74]

It also includes a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license

> *to the project manager and to recipients of software distributed by the project manager to make, have made, use, offer to sell, sell, important, and otherwise transfer the work, where such license applies only to those patent claims licensable by you that are necessarily infringed by your contribution(s)alone or by combination of your contribution(s) with the work to which such contribution(s) was submitted*

However, as stated in the Apache 2.0 license, the patent license will be terminated if

> *any entity institutes patent litigation against the contributor or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that the respective contribution or the work to which the contributor has contributed, constitutes direct or contributory patent infringement.*[75]

By using the Apache 2.0 license, the ETSI Open Source MANO initiative differentiates itself from the licensing approaches taken by the OpenAirInterface Alliance and the Broadband Forum. The latter two have drafted their own licenses, based on the Apache 2.0 license or based on the BSD-3 clause license and carved out an explicit patent grant referring to their respective FRAND licensing policies. OSM seems to be the first initiative to choose the Apache 2.0 itself and in its original version accepted by the FSF and OSI as conforming to their respective definitions as the only software license to be applied for software contributed to or developed in the context of the MANO software stack.

4.3 | Community perception

One of the main challenges for the FOSS ecosystem is license proliferation. The phenomenon of continued creation for new software projects or packages comes with the burden of an increasingly complex license interaction and license compatibility question. Consequently, drafting new licenses such as the OAI Public License 1.1 has encountered continued concerns and criticism from the FOSS community.

Particularly, the OAI Alliance has been criticized for requiring a copyright assignment from any contributor (Welte, 2016). One prominent voice in the FOSS community has summarized the OIA/OSA invitation to collaborate and contribute as "not very attractive" for the FOSS community:

> *It might all be well and fine for large industry and research institutes. But I do not think the Free Software Community has much to gain in all of this.*

In addition, the OAI public license 1.1 according to this commentator does

> *not fulfill the criteria of the FSF Free Software Definition as the license does not fulfill the freedom to run the program as you wish, for any purpose (including commercial use)*

and neither the criteria of the OSI initiative's Open Source Definition,

> *as the license must not discriminate against any fields of endeavor, such as commercial use (Welte, 2016).*

Perhaps even more critical to the OSI than license proliferation is the discussion surrounding the nomenclature and what exactly can and should be called "free software" and "open source software." Calling something "open" or "free" that is not aligned with the understanding developed by the FOSS community has always provoked a great deal of criticism.[76] In response to the recent discussions, a couple of statements have been published replicating the definitions of "Free Software" and "Open Source Software" as devised by the FSF or the OSI and calling for clarification that neither OAI Public License 1.1 nor the revised version of the BSD-clause as used by the Broadband Forum qualify as being free or open source software (Welte, 2016).[77] The OSI itself has published a statement addressing the question of openness through and in open source licenses.[78] This statement maintains that

> *the global software development and deployment community refers to software as "open source" when it is made available with source code under an OSI-approved IP license conveying the rights necessary to use, improve and share the software in a manner a given community considers appropriate.*[79]

Neither the OAI Alliance nor the Broadband Forum are explicitly mentioned in the OSI statement, but the publication date of October 2017 suggests a timely relevance.

4.4 | Analysis and policy implications

Analyzing the intersections of SEPs and FOSS goes far beyond the legal language of different licensing regimes and the question of license compatibility. Reflecting on the intersection of SEPs and FOSS development by analyzing the legal language of FRAND terms and FOSS licenses does not capture the actual situation. In order to understand the issues and discuss possible ways forward, the analysis provided above has to be put into context.

Both licensing regimes, FRAND-based licenses and FOSS licenses, have their own history and reasons for coming into existence. For FRAND commitments and related licensing terms, the goal is to provide a fair balance between the need for a fair return on investment to incentivize research and development as well as innovation *vis-a-vis* the need for broad availability and adoption of new technologies (European Commission Communication, 2016). For FOSS licenses, the original goal is best described in one of the earliest definitions published by the Free Software Foundation:

> *Free Software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software.*[80]

With regard to FRAND licensing terms, the goal of a fair balance between a fair return on investment and wide availability of new technologies gives flexibility in practice. Especially the fact that FRAND-based licensing terms are not standardized but individually negotiated between the respective patent owner and the implementer gives room for a broader and more differentiated interpretation of FRAND terms, including a royalty-free option for one specific group of implementers as long as neither party is treated detrimentally compared to any other similarly situated party.[81] In fact, the actual situation of the particular licensee seeking to rely on FRAND licensing terms can determine the understanding and definition of non-discrimination (Picht, GRUR Int. 2017, p. 572).

Interpreting FOSS licensing terms comes with less flexibility. First, FOSS licenses are standardized licensing terms designed to be used and adopted by many different licensors in order to reduce friction and transaction costs in the FOSS ecosystem. Secondly, all FOSS licensing terms are designed to ensure the licensee has the ability to study, change and further distribute the respective software. The latter point sets very specific framework conditions for the interpretation of FOSS licenses. Looking at the history of the FOSS development model, the FSF and OSI definitions play a critical role when it comes to the question of interpreting different terms of FOSS licenses. Of particular importance is the OSI definition, which has made its impact on the software market over recent years. Among the different criteria, the following requirement is particularly relevant with regard to SEP and possible FOSS implementations:

> *The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.*[82]

As outlined above, the legal language of most FOSS licenses facilitates the possibility to obtain individual patent licenses from the SEP owner.[83] However, the wording of the OSI definition clearly conflicts with these findings. According to the OSI definition, all downstream users, whether commercial or not, must be able to study, modify and further distribute the software without the need for execution of an additional license. Consequently, even if the legal language of most FOSS licenses can be read as being compatible with third-party patent licenses, this particular interpretation creates a conflict with the OSI definition and should therefore not be designated as such.[84] More precisely, FOSS is not defined only by license. Both definitions mentioned above, the definition cultivated by the Free Software Foundation and the definition maintained by the Open Source Initiative (Section 2.2 of this article), apply to software and can be used to evaluate whether a specific license can be adopted for FOSS. Even if many FOSS licenses are compatible with FRAND licensing terms, the concept of third-party patent licenses, including FRAND-based licenses, is not compatible with the OSI definition (in particular section 7 of the OSI definition). And even if the term "open source" is descriptive and not a registered trademark, calling software “open source” when it does not comply with the OSI definition will result in misleading declarations as to the rights attached to it and therefore give rise to competition concerns.[85]

Historically, the goal of both the free software movement and the open source software approach was to make software available under legal terms that allow and facilitate studying, modifying, and distributing the software without further restrictions.[86] While the term “open source software” was created as a strategic label to promote the idea of free software to business people and was successfully established on the market defined as such, it has preserved the initial objective of *free* software as first outlined in the 1970s and early 1980s.[87] Against this background, using copyright in software to be licensed under any type of FOSS license while retaining patents possibly infringed by contributions would be inconsistent with the original meaning of the term "open source".

Nevertheless, the history of FOSS development also reveals that there is a long tradition and willingness to work with commercial interests and to foster a healthy and competitive ecosystem around the development of FOSS. In fact, the discussion should not get stuck in the legal debate about licensing structures and related interpretation of licensing terms but rather focus on the ecosystem and the potential benefits for the end-users of future technologies. To enable a constructive dialogue and suggest possible solutions, practical issues of FOSS implementations should be identified and addressed, whereas ideological arguments should be neglected.

By way of example, accessibility and access conditions for organizations wishing to implement standards in software should be scrutinized. As outlined above, SDO policies are drafted to ask the owner of SEPs to be prepared to grant FRAND licenses to all implementers.[88] Hence, the detailed arrangements of the respective patent licenses—and especially the question of royalties—are left to the parties concerned, as these arrangements may differ from case to case. In other words, the SDOs are in no way involved in such license negotiations and it even seems questionable if and to what extent the commitment to be prepared to grant FRAND licenses is enforceable. Consequently, any organization wishing to implement a standard will first have to determine whether the implementation will in fact infringe any patent, then identify the respective patent owner, and after that negotiate individual (FRAND) licensing terms, potentially including compatibility issues with other already existing licensing obligations such as arising from some of the FOSS licenses analyzed herein. Given the intricacy of today's standards and the risk that not even all patents relevant to implement the standard are disclosed in the respective patent database, any implementer and especially those who do not have access to legal counsel will face high entry barriers.[89] While larger organizations, by reason of their greater resources, have several options to deal with patent owners by way of cross-licensing their own patents, respective patent pool arrangements, or simply accepting the risk of infringement and being prepared to defend or seek to invalidate the patent, smaller organizations may have to opt out (Lundell, Gamalielsson, & Katz, 2015, p. 62).

More transparency, supported and facilitated by the relevant SDOs, can help to address this problem of accessibility. Well managed, especially regularly maintained, and easily accessible patent databases will be the "low-hanging-fruit" to be tackled as a first step. At the same time, the FOSS community is invited to consider joint efforts with regard to legal support. Similarly to the work successfully done by different organizations in the context of license defense and litigation support, legal and strategic support with regard to standard implementation and SEP licensing compatibility or possible license negotiations could be initiated to support FOSS implementations.

Where FRAND-based licensing terms cannot be considered to include a royalty-free licensing rate, engaging in joint activities and pooling efforts and resources will be a possible solution. The Open Invention Network (OIN) has successfully demonstrated that there is a way to help FOSS companies and projects to overcome the challenges of patent law suits that is compatible with the culture of FOSS development (Nicholson, 2012, p. 16). OIN was initiated as a defensive patent pool to support FOSS projects and companies, particularly those associated with Linux. Backed by its core members, the goal was to create a collaborative ecosystem that enables companies to make significant corporate and capital expenditure investments in Linux—helping them to fuel economic growth. Over the past years, OIN has effectively supported Linux-based projects and businesses defeat or deflect the threat of patent litigation.[90] Similar efforts and initiatives should be explored in the context of standards to facilitate affordable and viable ways of FOSS implementations. Initiating a network that helps its members negotiate potential FRAND-based licensing terms could be one possible way forward. Differentiating between commercial implementations and non-commercial FOSS community implementations could be another. As outlined above, the definition of non-discriminatory can indeed accommodate different situations relevant for different licensees seeking to rely on FRAND licensing terms. As long as neither side is treated detrimentally compared to any other similarly situated party and to the extent that an objective reason exists, different licensing terms can be negotiated for different implementers within the definition of FRAND terms. The FOSS development model is characterized by way of standardized and royalty-free licensing terms, which provides a unique distinctive feature in relation to proprietary software development models and could be seen as an objective reason to apply a different set of FRAND licensing terms compared to commercial implementations.

Another possible way to stimulate further thinking could be a *pro rata* calculation. As long as the average selling price of the licensed products is zero, any percentage to be calculated as FRAND-based royalties will still be zero. In other words, only FOSS implementations that are commercially sold would be subject to a *pro rata* FRAND-based license calculation, whereas FOSS implementations that are used for non-commercial purposes would be licensed on a royalty-free FRAND based license. Instead of differentiating and modifying the patent grant depending on the

purpose of the licensee, as facilitated by the OpenAirInterface Software Alliance, a *pro rata* calculation of FRAND-based royalties would add consistency and transparency to the FRAND-based licensing process.

All possibilities mentioned above require further in-depth analysis and—most importantly—need to be facilitated in an open debate with all stakeholders participating. As long as the debate is mainly driven by ideological arguments, including critics of patent holdup, it will be difficult to enable a constructive and well-informed discussion and to identify practical means of collaboration. Staying with the question of patent holdup as an example,[91] relevant concerns are not unique to the intersection of SEPs and FOSS and not even unique to the standard setting process. The general question of whether royalties are too high and whether patent owners potentially abuse their market position should be discussed and analyzed independently from the intersection of SEPs and FOSS. Disproportionate behavior driven by self-interested motivations is a phenomenon that can not only be found in the context of patent ownership. Those who implement standards also have an interest in increasing profits. Patent royalties and especially the debate whether FRAND licenses should include zero royalties to allow for compatibility with FOSS licenses provide a welcome tool to reduce costs and thereby pursue economic self-interest. Hence, the debate about the intersection of SEPs and FOSS and interoperability between FRAND terms and FOSS licenses is not so much a problem of legal language in the respective licenses but a problem of conflicting interests and respective governance (Kappos, 2017, p. 265).

5 | CONCLUSION AND OUTLOOK

The analysis of the legal language of some of the most popular FOSS licenses reveals that there is no inherent legal conflict between FOSS and FRAND licensing models. Only very few FOSS licenses, including the original Copyleft license, the GPL, designed by the founder of the Free Software Foundation, do conflict with FRAND principles. The legal language of many popular permissive licenses such as the Berkeley Software Distribution (BSD) license, the MIT license or the ISC license are legally compatible with the FRAND licensing model (Kappos, 2017, p. 264). However, FOSS licenses cannot only be analyzed according to their legal language and literal interpretation but must be read in the context of the Free Software and Open Source definitions. The Free Software definition as well as the Open Source definition facilitated by the Open Source Initiative clearly require

> *the rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.*

The critical point to be discussed in the context of FOSS and FRAND compatibility is therefore not the legal language of individual FOSS licenses but the general definition of FOSS and the question of how FOSS will be defined and governed in the future. Even if the current definitions of FOSS create a conflict with the current setting of SEP's and related FRAND-based licensing terms, the history and analysis reveal that both FRAND-based licensing models and FOSS licensing models share a common goal—access to the core of the technology (Bird, 1998, pp. 76–80; Fogel, 2016). Hence, the problems commonly associated with SEPs and FOSS should not be seen as a conflict between documents and legal language but should be addressed by way of constructive collaboration. Most importantly, the discussion should not result or even end in the conclusion that some FOSS licenses are compatible with FRAND licensing, while others are not, and that drafting new licenses could fill the gap.

The question of to which extent a related governance model can and should be facilitated by SDOs and how the FOSS community can be involved is a very important aspect in the debate and should be addressed in a separate study. It would be useful to interview different stakeholders focusing on successful case studies and analyzing unsuccessful ones in order to further encourage constructive interaction. The result of such interviews and case studies is expected to result in the categorization of different scenarios, in which the intersection of SEPs and FOSS will play different roles. Consequently, the discussion surrounding a possible governance model may differ according to the different categories of interaction between SEPs and FOSS.

ABOUT THE AUTHOR

Dr. Catharina Maracke is a faculty associate at the Berkman Klein Center, Harvard University; a project associate professor at Keio University and a project associate professor at Kuehne Logistics University. Previously, Catharina served as international director at Creative Commons and was a member of the Global Agenda Council on the Intellectual Property System at the World Economic Forum. Her current work and interests include standards and intellectual property, compliance and the general interaction between law, technology and ethics. The author wishes to thank Sven Vetter, Humboldt-Universität zu Berlin, for his helpful comments and contributions to this research project.

ENDNOTES

[1] The term "Free and Open Source Software" includes both, Free Software as defined by the Free Software Foundation and Open Source Software as defined by the Open Source Initiative. In the following article, the term Free and Open Source Software (FOSS) will be used.

[2] See, Kappos (2017), p. 259 using Pokemon Go as an example.

[3] Communication from the Commission to the Institutions on Setting out the EU approach to Standard Essential Patents, retrieved from https://ec.europa.eu/docsroom/documents/26583, p. 12. In an earlier Communication "Towards interoperability for European public services," the European Commission even mentioned that FRAND licensing models should be compatible with FOSS licenses. The Communication calls for an "open specification" and describes its approach as follows: Intellectual Property Rights related to the specification are licensed under FRAND terms or on a royalty-free basis in a way that allows implementation in both proprietary and open source software. See https://ec.europa.eu/isa2/library/communication-towards-interoperability-european-public-services_en

[4] For details see Lundell and Gamalielsson (2017) and Lundell et al. (2015).

[5] See overview of software history at IBM, retrieved from https://www-03.ibm.com/ibm/history/history/decade_1960.html.

[6] See WEF Council on the Future of the Intellectual Property System: Software and Lessons from Free and Open Source Models in Intellectual Property Rights for the Global Creative Economy, retrieved from http://www3.weforum.org/docs/GAC/2013/WEF_GAC_IntellectualPropertyRights_GlobalCreativeEconomy_Report_2013.pdf.

[7] See WEF Council on the Future of the Intellectual Property System: Software and Lessons from Free and Open Source Models in Intellectual Property Rights for the Global Creative Economy, retrieved from http://www3.weforum.org/docs/GAC/2013/WEF_GAC_IntellectualPropertyRights_GlobalCreativeEconomy_Report_2013.pdf.

[8] See overview of history Fogel (2016), Producing Open Source Software—How to Run a Successful Free Software Project, O'Reilly Media, retrieved from https://producingoss.com/en/producingoss.pdf

[9] The GNU project, whereby GNU is a recursive acronym for "GNU is not Unix" hinting at the Unix Operating system, originally developed at the Bell Labs Research Center, written in C-Programming language: https://www.gnu.org/.

[10] See the GNU project's philosophy at https://www.gnu.org/philosophy/free-sw.en.html

[11] See Pragmatic Idealism, retrieved from https://www.gnu.org/philosophy/pragmatic.html.

[12] See also overview of categories of free and non-free software, retrieved from https://www.gnu.org/philosophy/categories.html#Non-CopyleftedFreeSoftware.

[13] See http://producingoss.com/en/introduction.html#free-vs-open-source.

[14] In response, the Free Software Foundation came up with a rather casual standard formula: "It's free as in freedom—think of it as in "free speech" and not in "free beer," retrieved from https://www.gnu.org/philosophy/free-sw.en.html.

[15] See Fogel (2016), Producing Open Source Software—How to Run a Successful Free Software Project, O'Reilly Media, retrieved from http://producingoss.com/en/introduction.html#free-vs-open-source, who described quality and profitability for certain businesses in certain circumstances as "merely pleasant side effects of a motive that was, at its root, neither technical or mercantile, but moral."

[16] For details see Jaeger and Metzger (2016), Open Source Software, 4th edition, p. 9 et *seq.* In particular, many leaders and developers of the most important free and open source software projects voted for the term "Open Source" software in an event hosted by technology publisher Tim O'Reilly in April 1998. See history of OSI, archive retrieved from https://web.archive.org/web/20021001164015/ http://www.opensource.org/docs/history.php.

[17] See Bruce Perens in an interview with The Register in 2018 https://www.theregister.co.uk/2018/02/03/open_source_turns_20/).

[18] Supporters of the term open source formed the Open Source Initiative as organization that could serve as a steward to the Open Source definition and—even more importantly—as a community-recognized body for reviewing and approving licenses as conformant with the open source definition, see https://opensource.org/about.

[19] For details, see Open Source Definition, originally derived from the Debian Free Software Guidelines, retrieved from: https://opensource.org/osd.

[20] In the interest of readability and simplification, this article will use the term FOSS as an abbreviation to refer to both, Free Software and Open Source Software.

[21] See also detailed explanation, retrieved from https://www.gnu.org/licenses/copyleft.en.html.

[22] Taking the LGPL as a prominent example of a weak copyleft license, see Jaeger and Metzger (2016), Open Source Software, 4th edition 2016, p. 95 et. seq.

[23] See section 3 of the Apache 2.0 license and section 13 of the Perl Artistic license 2.0, including a very similar termination clause.

[24] See section 2 of the EUPL license 1.2 without any termination clause.

[25] For the U.S. perspective Nadan, Closing the Loophole: Open Source Licensing & The Implied Patent License, *The Computer & Internet Lawyer*, *Vol. 26 No. 8 August 2009*, pp. 1–6.

[26] Nadan, Closing the Loophole: Open Source Licensing & the Implied Patent License. *The Computer & Internet Lawyer*, *Vol. 26 No. 8 August 2009*, p. 2 referring to *State Contracting & Engineering Corp*. v. Florida 258 E3d 1329, 1339 (Fed. Cir. 2001), cert. denied, 534 U.S. 1131 (2002).

[27] State Street Bank & Trust Co *vs.* Signature Financial Group Inc., 149 F.3d1368 (Fed. Cir. 1998) was the first Federal Circuit's decision explicitly recognizing the patentability of software. In an earlier decision, the Supreme Court questioned whether a software enabled invention in the form of an algorithmic process could be patentable, but ruled that a process claim directed to a numerical algorithm, as such, was not patentable because "the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself." See Gottshalk v. Benson, 409 US 63 (US Supreme Court 1972).

[28] See https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html.

[29] See https://www.apache.org/licenses/LICENSE-2.0.

[30] See http://www.perlfoundation.org/artistic_license_2_0.

[31] See https://opensource.org/licenses/EUPL-1.1.

[32] See https://www.cencenelec.eu/standards/DefEN/Pages/default.aspx.

[33] See Communication from the Commission to the Institutions on Setting out the EU approach to Standard Essential Patents, retrieved from https://ec.europa.eu/docsroom/documents/26583 referring to a McKinsey Global Institute (2015) study that predicts the IoT sector to grow up to nine trillion Euro's per year in developed countries by 2025.

[34] Without interoperability, enabled by standards, 40% of the potential benefits of the IoT systems would not be reaped, see Communication from the Commission to the Institutions on Setting out the EU approach to Standard Essential Patents, retrieved from https://ec.europa.eu/docsroom/documents/26583 p. 1.

[35] Gupta (2018) describes the Creation of Technical Standards "Unpacking the Mobile Industry's 3GPP Standards."

[36] See Ménière, Science and Policy Report by the EC Joint Research Centre, Fair, Reasonable and Non-Discriminatory (FRAND) Licensing Terms (2015), p. 7, who mentions Adobe's PDF computer file format and Microsoft's Office Open XML format as well-known examples of *de facto* standards.

[37] See Contrereas (2018), in The Cambridge Handbook of Technical Standardization Law (J. L. Contreras ed.), Chapter 9, pp. 149–169, who provides a detailed view on the "Origins of FRAND Licensing Commitments in the United States and Europe." See also p. 210.

[38] See also Contrereas (2018), in The Cambridge Handbook of Technical Standardization Law (J. L. Contreras ed.), p. 163.

[39] For details see Contreras, in The Cambridge Handbook of Technical Standardization Law (J. L. Contreras ed. 2018), p. 169. Shortly after ETSI was founded in 1988, he European Commission published its Communication on Intellectual Property Rights and Standardization (1992) and encouraged ETSI to revise its approach and policies a couple of times before today's version came in place.

[40] Lemley (2002), *90 Cal. L. Rev. 1889*, provides an *Appendix* with "Standard-Setting Organization Intellectual Property Policies" which shows that licensing provisions including "RAND" were already quite common at the beginning of the Twenty-First Century.

[41] See CJEU Case C-170/13–Huawei/ZTE.

[42] See the review of the recent approach of the UK Patent Court by Picht, *GRUR Int*. 2017, p. 569.

[43] See Nilsson, *2017*, p. 1017 et *seq*. and also Goldstein & Kearsey (2004), Technology Patent Licensing: An International Reference on 21st Century, US: Aspatore Inc., pp. 26–28 on the aspect of being "reasonable."

[44] For an overview of the debate about software in general, see Guadamuz (2006), The Software Patent Debate, *Journal of Intellectual Property Law and Practice*, p. 2 et. seq.

[45] When hosting a FOSS project in a standardization process, possible questions include whether and how it can be ensured that all contributed software is FOSS and contributed patents are equally and openly licensed.

[46] See different cases if potential interaction structured and evaluated by Lundell & Gamalielsson (2017), On the potential for improved standardization through use of open source work practices in different standardization organizations: How can open source projects contribute to development of IT standards? EURAS Proceedings, pp. 137–155.

[47] "Software is Eating the World" as famously stated by Marc Andreesen in 2011, retrieved from https://a16z.com/2016/08/20/why-software-is-eating-the-world/.

[48] See 2015 study, retrieved from http://www.zdnet.com/article/its-an-open-source-world-78-percent-of-companies-run-open-source-software/; a related study shows that 97% of the top 500 supercomputers run on linux, data retrieved from https://itsfoss.com/97-percent-worlds-top-500-supercomputers-run-linux/ and for more general data see OpenSource Survey at GitHub (2017), retrieved from http://opensourcesurvey.org/2017/.

[49] Details of this argument are set out in several statements and papers, including Stallman (2002), Patent Licenses Discriminate, retrieved from http://www.zdnet.com/article/stallman-patent-licenses-discriminate/

[50] Morgan (2002), Open Source Software and Software Patents: Finding the Common Ground in a Patent Pool, retrieved from https://flosshub.org/system/files/morgan.pdf, who claims that it would be "virtually impossible to meet this restriction if a patent holder requires the patent licensee to pay a royalty for each copy of the software distributed." For an overview of the arguments see also Kesan (2011), The Fallacy of OSS Discrimination by FRAND Licensing: An Empirical Analysis, *Illinois Public Law and Legal Theory Research Paper Series No. 10–14*, p. 3.

[51] IEEE specified the smallest saleable unit principle (SSUP), which requires that a reasonable royalty calculation should consider the value of the smallest saleable compliant implementation that practices a patent claim, which has been discussed controversially, see Teece and Sherry (2016), The New IEEE IPR Policy: Did the IEEE Shot Itself in the Foot and Harm Innovation? retrieved from http://businessinnovation.berkeley.edu/wp-content/uploads/2014/07/14-The-IEEEs-New-..Policy_Teece_Sherry_8-3-16_2_Clean.pdf

[52] Mitchell and Mason (2011), Compatibility Of The Licensing Of Embedded Patents with Open Source Licensing Terms, *International Free and Open Source Law Review*, *Vol. 3 Nr. 1*, p. 6, who consider this interpretation referring to Section 0 of the GPL that explicitly talks about "program or any other derivative work under copyright law."

[53] See overview at BlackDuck Software at https://www.blackducksoftware.com/top-open-source-licenses

[54] See https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html

[55] See B. 3.

[56] See https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html

[57] See D. a.

[58] See C. a.

[59] See line of argumentation at Mitchell and Mason (2011), Compatibility Of The Licensing Of Embedded Patents with Open Source Licensing Terms, *International Free and Open Source Law Review*, *Vol. 3 Nr. 1*, p. 6.

[60] See also details at Haedicke and Timmann (2014) p. 303 elaborating the theory of transfer tied to purpose (Zweckuebertragungslehre).

[61] See B. 3.

[62] See https://opensource.org/licenses/MIT

[63] See http://www.openairinterface.org/?page_id = 72

[64] See the Institut Mines-Télécom's (2017) interview with Christian Bonnet, retrieved from https://blogrecherche.wp.imt.fr/en/2017/05/23/openairinterface-5g-system-of-future/

[65] OSA currently provides a standard-compliant implementation of a subset of Release 10 LTE for UE, eNB, MME, HSS, SGw and PGw on standard Linux-based computing equipment (Intel x86 PC/ARM architectures). The OAI Public License is retrieved from http://www.openairinterface.org/?page_id = 698.

[66] Explanation retrieved from http://www.openairinterface.org/?page_id = 101.

[67] The Broadband Forum serves as the successor of the DSL Forum, which was founded in 1996.

[68] Section 3.1 of the IPR Policy retrieved from https://www.broadband-forum.org/IPRPolicy2016.pdf.

[69] Section 4.3 of the IPR Policy retrieved from https://www.broadband-forum.org/IPRPolicy2016.pdf.

[70] Section 4.3 of the IPR Policy retrieved from https://www.broadband-forum.org/IPRPolicy2016.pdf.

[71] For a similar approach, see the license text of the BSD 3-Clause Clear License, retrieved from https://spdx.org/licenses/BSD-3-Clause-Clear.html, which uses explicit Copyright terminology.

[72] See http://www.etsi.org/technologies-clusters/technologies/nfv/open-source-mano.

[73] Open Source Mano Terms of Reference, retrieved from https://osm.etsi.org/about/osm-license.

[74] Section 2 of the ETSI OSG OSM Corporate Contributor License Agreement, retrieved from https://portal.etsi.org/Portals/0/TBpages/OSM/Docs/ETSI_OSG_OSM_Member_Agreement_2016-02-09.pdf.

[75] Section 3 of the ETSI OSG OSM Corporate Contributor License Agreement, retrieved from https://portal.etsi.org/Portals/0/TBpages/OSM/Docs/ETSI_OSG_OSM_Member_Agreement_2016-02-09.pdf.

[76] See FAQ at the Open Source Initiative, retrieved from https://opensource.org/faq#avoid-unapproved-licenses, Peter Suber on Open Access, SPARC Open Access Newsletter, September 2, 2004, Creative Commons on Open Educational Resources (OER), retrieved from https://www.cccoer.org/learn/open-licensing/. retrieved from https://dash.harvard.edu/bitstream/handle/1/3997159/suber_news77.html.

[77] See Fair Standards Alliance (2017), The Importance of Maintaining the Open Source Software Value Proposition, retrieved from http://www.fair-standards.org/learn-more/.

[78] See https://opensource.org/node/906.

[79] See https://opensource.org/node/906.

[80] See definition and details at https://www.gnu.org/philosophy/free-sw.en.html.

[81] See above section 3.2.

[82] No. 7 of the OSI Definition, retrieved from https://opensource.org/osd.

[83] See section 4.1.3.

[84] According to the Open Source definition, any licensee of FOSS can expect to use the respective software "without the need for execution of an additional license by those parties," retrieved from https://opensource.org/osd. It would therefore be misleading to receive software under a pretended FOSS license whereby the licensee would still need an additional patent license and pay royalties for it.

[85] Since the term Open Source has been successfully established on the market as representing software licensed under very specific criteria and maintained by the Open Source initiative, using the term for software licenses that don't comply with the Open Source criteria would lead to competition concerns in the software market. For details on competition law see Jaeger & Metzger (2016) p. 309–328, Köhler, Bornkamm & Feddersen (2018) § 5 n. 1.1 et seq. and Ohly & Sosnitza (2016) § 5 n. 102 et seq.

[86] See section 2.2.

[87] See above section 2.1.

[88] See Section 6 of the current ETSI IPR Policy as an example: http://www.etsi.org/images/files/IPR/etsi-ipr-policy.pdf.

[89] According to Baron and Pohlmann (2018), more than 100.000 SEPs have been declared under FRAND commitments within the ETSI framework, see Baron and Pohlmann (2018), Mapping Standards to Patents using Databases of Declared Standard-Essential Patents and Systems of Technological Classification," retrieved from https://pdfs.semanticscholar.org/f65a/14f24ab67ba48aeedd6bd29a6957818716a1.pdf.

[90] For details see Nicholson (2012), Open Invention Network: A Defensive Patent Pool for Open Source Projects and Businesses, Technology Innovation Management Review, p. 12–17.

[91] Simply put, patent holdup describes the concern that patent holders may be able to exploit the cooperative standard setting process to extract excessive and disproportional royalties from standards implementers, see Simcoe and Shampine (2018), Economics of Patents and Standardization: Network Effects, Hold-up, Holdout, Stacking, in Contreras, in The Cambridge Handbook of Technical Standardization Law (J. L. Contreras ed.), p. 104.

ORCID

Catharina Maracke http://orcid.org/0000-0003-2626-8729

REFERENCES

Baron, J. & Pohlmann, T. (2018). Mapping standards to patents using databases of declared standard-essential patents and systems of technological classification. Available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3119145 (accessed in June 2018).

Bird, G. (1998). The business benefit of standards. *StandardView*, *6*(2), 76–80. https://doi.org/10.1145/301688.301691

Black Duck & North Bridge. (2015). Future of open source. Available at http://www.zdnet.com/article/its-an-open-source-world-78-percent-of-companies-run-open-source-software/ (accessed in June 2018).

Blind, K., & Kahin, B. (2018). Standards and the global economy. In J. L. Contreras, (Ed.), *The cambridge handbook of technical standardization law* (pp. 7–16). Cambridge: Cambridge University Press. Available at: https://doi.org/10.1017/9781316416723.003 (accessed in June 2018).

Broadband Forum. (2016). Intellectual Property Rights Policy. Available at: https://www.broadband-forum.org/IPRPolicy2016.pdf (accessed in June 2018).

Contrereas, J. (2018). In J. L. Contreras, (Ed.), *The cambridge handbook of technical standardization law* (pp. 149–169). Cambridge: Cambridge University Press.

Dapp, T., & Bernauer, M. (2009). Hot debate about chilling effects: Do software patterns hamper/Free Open Source Software Development? CIS Working Paper, published by the Center for Comparative and International Studies (ETH Zurich and University of Zurich) No. 40 (2009), pp. 1–26. Available at: https://poseidon01.ssrn.com/delivery.php?ID=349115115064091012094094124092092064046063048046095065028125072121067002023004028108103013042047060122055023122088024013119004116054021030082000080073123116120071101086055022027114010110102096079085076065026102064119120127009104076087123111113087088126&EXT = pdf (accessed in June 2018).

European Commission. (2016). Communication, ICT Standardization Priorities for the Digital Single Market, April 19th 2016. Available at https://ec.europa.eu/digital-single-market/en/news/communication-ict-standardisation-priorities-digital-single-market (accessed in June 2018).

European Commission. (2017). Communication from the Commission to the Institutions on Setting out the EU approach to Standard Essential Patents. Available at https://ec.europa.eu/docsroom/documents/26583 (accessed in June 2018).

Fair Standards Alliance (2017). The importance of maintaining the Open Source Software Value proposition. Available at http://www.fair-standards.org/learn-more/ (accessed in June 2018).

Fitzgerald, A., & Pappalardo, K. (2009). Moving towards open standards. *SCRIPted*, *6*(2), 467–483. https://doi.org/10.2966/scrip.060209.467. Available at: http://eprints.qut.edu.au/30118/ (accessed in June 2018).

Fogel, K. (2016). Producing Open Source Software—How to Run a Successful Free Software Project, O'Reilly Media. Available at https://producingoss.com/en/producingoss.pdf (accessed in June 2018).

Free Software Foundation. What is free software? The Free Software Definition. GNU operating system. Available at https://www.gnu.org/philosophy/free-sw.en.html (accessed in June 2018).

Goldstein, L., & Kearsey, B. (2004). Technology patent licensing: An international reference on 21st century. US: Aspatore Inc.

Guadamuz, A. (2006). The software patent debate. *Journal of Intellectual Property Law and Practice*, *2006*, 1–11. https://pdfs.semanticscholar.org/2cf6/c746fa8a4228b885b3f32600896a90f780af.pdf (accessed in June 2018).

Gupta, K. (2018). In J. L. Contreras, (Ed.), *The cambridge handbook of technical standardization law* (pp. 31–42). Cambridge: Cambridge University Press.

Haedicke, M., & Timmann, H. (2014). *Patent law handbook. European and German patent law*. Munich: Beck.

IBM. Chronological History of IBM. 1960. Available at: https://www-03.ibm.com/ibm/history/history/decade_1960.html (accessed in June 2018).

Institut Mines-Télécom. (2017). Interview with Christian Bonnet, Available at https://blogrecherche.wp.imt.fr/en/2017/05/23/openairinterface-5g-system-of-future/ (accessed in June 2018).

Jaeger, T., & Metzger, A. (2016). *Open Source Software*. 4th edition. Munich: Beck.

Kappos, D. J. (2017). Open source software and standards development organizations: symbiotic functions in the innovation equation. *Columbia Science & Technology Law Review*, *XVIII*, 259–268. http://www.stlr.org/download/volumes/volume18/kappos.pdf (accessed in June 2018).

Kappos, D. & Harrington, M. Y. (2018). The truth about OSS-FRAND: By all indications, compatible models in standards settings. The Columbia Science and Technology Law Review XIX. Available at: http://stlr.org/2018/10/15/the-truth-about-oss-frand-by-all-indications-compatible-models-in-standards-settings/ (accessed in October 2018).

Kesan, J. P. (2011). The fallacy of OSS discrimination by FRAND licensing: An empirical analysis, Illinois Public Law and Legal Theory Research Paper Series No. 10–14. https://doi.org/10.2139/ssrn.1767083. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1767083 (accessed in June 2018).

Köhler, H., Bornkamm, J., & Feddersen, J. (2018). *Gesetz gegen den Unlauteren Wettbewerb* (36th ed), Munich: Beck.

Lemley, M. (2002). Intellectual property rights and standard-Setting organizations. 90 Cal. L. Rev. 1889, https://doi.org/10.15779/Z384D9P. Available at: https://scholarship.law.berkeley.edu/californialawreview/vol90/iss6/3/ (accessed in June 2018).

Lundell, B., & Gamalielsson, J. (2017). On the potential for improved standardization through use of open source work practices in different standardization organizations: How can open source projects contribute to development of IT standards? In K. Jakobs, & K. Blind, (Eds.), *Digitalisation: Challenge and opportunity for standardisation: Proceedings of the 22nd EURAS annual standardisation conference* (pp. 137–155). Aachen: Verlag Mainz.

Lundell, B., Gamalielsson, J., & Katz, A. (2015). Implementation of open standards in software: To what extent can ISO standards be implemented in open source software? *International Journal of Standardization Research*, *13*(1), 47–73. https://doi.org/10.4018/IJSR.2015010103. Available at: https://zdoc.site/on-implementation-of-open-standards-in-software.html (accessed in June 2018).

McKinsey Global Institute. (2015). The Internet of Things, Mapping the Value behind the Hype. Available at: https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/The%20Internet%20of%20Things%20The%20value%20of%20digitizing%20the%20physical%20world/The-Internet-of-things-Mapping-the-value-beyond-the-hype.ashx

Ménière, Y. (2015). Science and Policy Report by the EC Joint Research Centre, Fair, Reasonable and Non-Discriminatory (FRAND) Licensing Terms. DOI: 10.2791/348818. Available at: http://publications.jrc.ec.europa.eu/repository/bitstream/JRC96258/jrc96258.pdf (accessed in June 2018).

Mitchell, I. G., & Mason, S. (2011). Compatibility of the licensing of embedded patents with open source licensing terms. *International Free and Open Source Law Review*, *3*(1), 25–58. https://doi.org/10.5033/ifosslr.v3i1.57. Available at: http://www.ifosslr.org/ifosslr/article/view/57/99 (accessed in June 2018).

Morgan, J. (2002). Open source software and software patents: Finding the common ground in a patent pool. Available at: https://flosshub.org/sites/flosshub.org/files/morgan.pdf (accessed in June 2018).

Nadan, C. H. (2009). Closing the loophole: Open source licensing & the implied patent license. *The Computer & Internet Lawyer*, *26*(8), 1–6. Available at: https://de.scribd.com/document/46088081/Closing-the-Loophole-Open-Source-Licensing-amp-the-Implied-Patent-License-Nadan (accessed in June 2018).

Nicholson, D. (2012). Open invention network: A patent pool for open source projects and businesses. *Technology Innovation Management Review*, *2*(1), 12–17. Available at: https://timreview.ca/article/511 (accessed in June 2018).

Nilsson, F. (2017). Appropriate base to determine a fair return on investment: A legal and economic perspective on FRAND. GRUR Int. 2017, pp. 1017–1024.

Ohly, A. & Sosnitza K. (2016). *Gesetz gegen den unlauteren Wettebwerb* (7th ed), Munich: Beck.

OpenAirInterface. (2018). 5G software alliance for democratising wireless innovation. OAI Public License V1.1. Available at: http://www.openairinterface.org/?page_id = 698 (accessed in June 2018).

O'Reilly, T. (1998). History of OSI, archive available at https://web.archive.org/web/20021001164015/http://www.opensource.org/docs/history.php (accessed in June 2018).

Perens, B. (1999). The Open Source Definition in Open Sources: Voices from the Open Source Revolution. 1st edition. Available at http://www.oreilly.com/openbook/opensources/book/perens.html (accessed in June 2018).

Picht, P. (2017). Unwired Planet v. Huawei: A Seminal SEP/FRAND Decision from the UK. GRUR Int. 2017, p. 569–580.

Simcoe, T., & Shampine, A. (2018). Economics of patents and standardization: network effects, hold-up, holdout, stacking, in Contreras. In J. L. Contreras, (Ed.), *The cambridge handbook of technical standardization law* (pp. 100–124). Cambridge: Cambridge University Press. https://doi.org/10.1017/9781316416723.010. Available at: https://www.cambridge.org/core/books/cambridge-handbook-of-technical-standardization-law/economics-of-patents-and-standardization-network-effects-holdup-holdout-stacking/17500E183DC72E6C94847CB5042A06D9 (accessed in June 2018).

Stallman, R. (2002). Stallman: "Patent Licenses Discriminate." ZDNET. Available at: https://www.zdnet.com/article/stallman-patent-licenses-discriminate/ (accessed in June 2018).

Teece, D., & Sherry, E. (2016). The New IEEE IPR Policy: Did the IEEE shot itself in the foot and harm innovation? Available at https://businessinnovation.berkeley.edu/wp-content/uploads/2014/07/Tusher-Center-Working-Paper-No.-13.pdf (accessed in June 2018).

Welte, H. (2016). On the OpenAirInterface re-licensing. Available at: http://laforge.gnumonks.org/blog/20160201-oai-osa-licensing/ (accessed in June 2018).

World Economic Forum. (2013). Council on the Future of the intellectual property system: Software and lessons from Free and Open Source Models in Intellectual Property Rights for the Global Creative Economy. Available at http://www3.weforum.org/docs/GAC/2013/WEF_GAC_IntellectualPropertyRights_GlobalCreativeEconomy_Report_2013.pdf (accessed in June 2018).

How to cite this article: Maracke C. Free and Open Source Software and FRAND-based patent licenses. *J World Intellect Prop*. 2019;22:78–102. https://doi.org/10.1111/jwip.12109