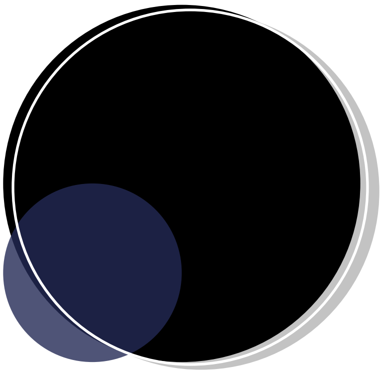

OPEN SOURCE AND FRAND: WHY LEGAL ISSUES ARE THE WRONG LENS

Simon Phipps

As patent-centric technology industries have realised the power of the open source approach used in the pure-software industry, they have increasingly sought to derive its benefits. In arguing their case, they have sought to characterise open source as a matter of licensing technicalities, and then aimed to identify ways to change the licensing to fit their unrelated business practices. This article explores the phenomenon, introduces the terms "requirements-led" and "implementation-led" to help distinguish the differing world views and suggests the legal lens is not only wrong but leads to failure.

ofa OpenForum Academy

2019
MARCH

OPEN SOURCE AND FRAND:
WHY LEGAL ISSUES ARE
THE WRONG LENS

ABOUT

OpenForum Europe (OFE) is a not-for-profit, independent European based think tank which focuses on openness within the IT sector. We draw our support not only from some of the most influential global industry players, but most importantly from across European SMEs and consumer organisations and the open community. OFE also hosts a global network of *OpenForum Academy Fellows*, each contributing significant innovative thought leadership on core topics. Views expressed by OFE do not necessarily reflect those held by all its supporters.

2019
MARCH

OPEN SOURCE AND FRAND: WHY LEGAL ISSUES ARE THE WRONG LENS

Simon Phipps[1]

As patent-centric technology industries have realised the power of the open source approach used in the pure-software industry, they have increasingly sought to derive its benefits. In arguing their case, they have sought to characterise open source as a matter of licensing technicalities, and then aimed to identify ways to change the licensing to fit their unrelated business practices. This article explores the phenomenon, introduces the terms "requirements-led" and "implementation-led" to help distinguish the differing world views and suggests the legal lens is not only wrong but leads to failure.

OVERVIEW

In considering the relationship between patent-bearing standards and open source implementations, the question many start with is "Are open source licenses compatible with FRAND?" But this is like asking if apples are compatible with potatoes; the two are different types of promises in different categories applicable to different needs. A FRAND promise applies to an entire specification, however implemented, and includes the possibility of further negotiations over royalty-based licensing.

The Open Source promise, on the other hand, is much more narrow. The scope of the promise is limited to specific software code, not an entire specification, and is automatic without the need for further negotiations. A company that makes a FRAND promise applicable to an entire specification (however implemented) can also make – or not make – an additional open source promise to license a particular software-based implementation of that specification. Companies, in fact, do so with regularity.

[1] Simon Phipps is president of the Open Source Initiative and director of the Open Rights Group and of The Document Foundation. He is managing director at Meshed Insights Ltd. He was previously head of open source at Sun Microsystems and a founder of the Java Technology Centre at IBM. Contact him at simon@meshedinsights.com. This article does not necessarily reflect the views of any organisation the author may represent or of any patron. © 2019 Meshed Insights Ltd, licensed CC-BY-SA."

But the real issue around successful interaction and collaboration between open source and standards is not legal. The devil is not in the legal details in this case; rather, the legal details may themselves prevent understanding of the real situation. This is an issue of collaboration between multiple stakeholders, so what ultimately matters is what the stakeholders believe, and whether they will choose to collaborate. Putting aside legal disputes about the meaning of open source licenses (including how they convey patent rights), the fact is that software projects seeking to apply the FRAND patent licensing practices customary in some standards development organizations will not be viewed as open source by most developers. This will not only dissuade collaboration, but also – if they are (incorrectly) referred to as open source projects – create dissatisfaction and ill will, undermining the potential synergies.

A PRACTICAL PROBLEM

The plain truth is that genuine open source communities – explained in more detail below -- do not form around code-bases where an arbitrary user needs to negotiate terms with a rights holder at any point in their cycle of adoption and improvement of the code. This is not a matter of "compatibility" in any sense that legal experts consider. The open source approach simply depends on all the rights necessary to use the source code for any purpose being granted in advance. A regimen where some rights are withheld subject to negotiation will at best be dysfunctional. The orthogonality of the rights-multilaterally-in-advance and rights-bilaterally-negotiated principles is the problem, not some nuance in the license; the absence of an emergent network effect is the failure mode, not illegality or incompatibility.

Attempting to analyse this dysfunction in terms of "legal compatibility" is thus misdirection. FRAND promises are different from open source not because of some kind of general and technical incompatibility with licenses[2], but because the FRAND promise includes the expectation that there will be multiple, negotiated, bilateral relationships

[2] As a matter of observation, OSI is traditionally sensitive to the views of a diverse range of interests so has acted to ensure they remain engaged, but it has never approved licenses that exclude patent rights. For example, see comments by OSD author Bruce Perens: http://lists.opensource.org/pipermail/license-review_lists.opensource.org/2009-April/000757.html, the discussion around CC0: https://opensource.org/faq#cc-zero and the content of the OSI Open Standards Requirement: https://opensource.org/osr. This is discussed in more detail in Lindberg, V "OSS and FRAND: Complementary Models for Innovation and Development" (2019), Columbia Science and Technology Law Review, http://stlr.org/2019/03/04/oss-and-frand-complementary-models-for-innovation-and-development/

between patent owners and code users. The open source regime does not include such side-tracks[3].

Rather, requiring one-by-one negotiations between patent owners and code users is at fundamental odds with the collaborative, co-creational method of open source that emerges when developers are assured of their freedoms[4] to innovate without worrying about seeking additional permission. You can implement FRAND-licensed standards in software and (usually incorrectly) call such software "open source", but almost no-one outside your grouping will collaborate with you or use your implementation, as has been repeatedlyy discovered[5].

How is that so? Open source licensing is multi-lateral and does not generally consider relationships that are conducted outside the scope of the license and community[6]. So a group with patent relationships negotiated outside the scope of the code-base can also develop implementations of a standard and license their software under certain OSI-approved licenses[7]. It is thus perfectly possible for a project to operate FRAND licensing under a number of open source licenses, but the result is that the contributors will convey to all recipients of the code an expectation of patent rights[8]. If companies that participate in the standard choose not to participate in the open source projects-

[3] In fact the Open Source Definition – a foundational document of the open source movement - expressly precludes requirements for royalties or separately-negotiated licenses, see https://opensource.org/osd-annotated especially clauses 1 and 7.

[4] See the Free Software Foundation's explanation of this concept https://www.gnu.org/philosophy/free-sw.en.html

[5] See for example Lundell, Gamalielsson and Katz "On Implementation of Open Standards in Software: To What Extent Can ISO Standards be Implemented in Open Source Software?" (2015), International Journal of Standardization Research Vol 13, No 1.

[6] The GPL family of licenses is an exception; it does not tolerate patent rights that only apply to some users. See sections 11 and 12 of the GPLv3 for example, https://www.gnu.org/licenses/gpl-3.0.en.html

[7] For more information about the Open Source Initiative's licensing approval role see https://opensource.org/licenses. OSI is a public charity established by some of the coiners of the term "open source" in 1998 and now represents more than 60 open source organisations globally as well as a growing number of individual members.

[8] Modern open source licenses include explicit patent grants, ensuring contributors all benefit from the same rights. There's good evidence that even licenses like BSD & MIT should be interpreted, including in European jurisdictions, as including a grant of patent rights due to the reasonable expectation of the recipient that they should. See Peterson's discussion of the matter under US law, https://opensource.com/article/18/3/patent-grant-mit-license where he proposes this can be even be considered an explicit rather than implicit grant. For example, there are reasonable grounds for a recipient of code under the MIT license to expect that <COPYRIGHT HOLDER> has granted them all the rights necessary "to deal in the Software without restriction", with no exclusions for selected rights classes. Lindberg (ibid.) also demonstrates Kappos is wrong to assert the intent of the authors of these licenses was to exclude patent rights.

so as to avoid such a result, that is of course always their option but it's likely to be self-defeating.

Why self-defeating? If a standards developer initiates a software project while seeking to reserve all patent rights, very few (if any) from the open source community will join in; rather, such outsiders may well start a competing activity instead[9]. This undermines the benefits of collaboration with open source – such as the widespread developer acceptance, the network-effect market making or the aim to develop faster, cheaper and with more diverse contribution and innovation. For these purposes, the negotiation-mandatory FRAND-based world will never deliver in an open source context. Developers will simply work around the barriers that have been erected, choosing other standards, building on other code-bases where their freedoms are guaranteed and collaborating in venues where the governance assures them all participants are equal and no-one is seeking to control them and their work. What's more the absence of that open, collaborative, equal environment that is the generally-understood meaning of the term-of-art "open source" will be found to be absent, leading many to believe there was an attempt to deceive.

ABOUT OPEN SOURCE

Why is this the case? Open source is a development of the earlier Free Software movement, which seeks to guarantee the rights of users of software to self determination by ensuring they can use, study, improve and share all the software that impacts their life, without needing to seek permission beyond the accompanying license[10]. Open source focusses on the exercise of those "four freedoms" to empower a collaborative methodology that leverages multiple, disconnected, extrinsic motivations for the use and thus for co-development of a body of software. OSI-approved open source licenses guarantee all rights necessary for this to work are available, to all, unconditionally, at the outset.

Open source does not primarily aim to create end-user works or to provide software without charge, although

[9] For example, the approach taken to the development of FRAND-based CODECs by members of the Motion Picture Expert Group led directly to the formation of the Alliance for Open Media and the emergence of AV1.

[10] See https://www.fsf.org/about/what-is-free-software for a full explanation.

those are common consequences. Rather, it seeks to share the development and maintenance of bodies of non-differentiating (though highly innovative) software to the common good of the participants in a community. It is renowned for creating powerful network effects that accelerate adoption of a software implementation and any standards it embodies.

The open source approach arose in a software industry where patents were essentially absent[11] and where standardisation typically followed rather than created markets (and often focussed on hardware). At the margins where the software touched media or semiconductor chips there have been some encumbered standards that have seeped in from other industries, but to the best approximation the world in which open source emerged has no standard-essential patents and its very nature is that such patents would not be an issue since all contributors will have granted all rights necessary to use the software through the associated license.

SEQUENCE OF INNOVATION

A source of confusion arises from the fact there are essentially two different models for standardisation in play. While they have much common terminology, the sequence of innovation and standardisation is very different and as a result people in each model frequently misunderstand the expectations of those from the other.

Requirement-led standards

A *requirement-led* model is adopted by industries where high degrees of interoperability are essential to market formation and operation. In this model (epitomised by the mobile phone market), an industry forum acts as the venue for deployers to specify the capabilities they need in some form of requirements document. Suppliers then propose solutions to these requirements and declare in advance any patent interest they may have in their proposals. The industry forum then eva-

[11] In fact, Bessen and Maskin argued a decade ago that the absence of patents may well have been essential to the industry; "For industries like software or computers, theory suggests that imitation may promote innovation and that strong patents (long-lived patents of broad scope) might actually inhibit it." See Bessen, J. and Maskin, E. (2009) Sequential innovation, patents, and imitation, RAND Journal of Economics, Vol. 40, No. 4, pp. 611-635.

luates proposals against the requirements and selects a fit at each function point to build a specification, accepting as "standard essential" the patents disclosed in the process. The specification is ratified as a standard and a market forms in which suppliers and deployers trade. The suppliers whose proposals were included in the standard are then free to recover their costs and generate reasonable profit from all implementors of the standard based on their "Standard-Essential Patents" (SEPs).

Implementation-led standards

An alternative, *implementation-led* is adopted largely in the pure software ("ICT") industry. In this model, an early market entrant or market leader innovates privately and brings a new product to market. The product will expose some form of external interface -– a file format, an API, a protocol, a language – and over time partners will implement new capabilities using that exposed interface. As the market emerges and competitors begin to re-implement the interface, demand arises for standardisation. An industry forum acts as the venue for various market stakeholders to discuss the canonical form of the interface, and a specification is collaboratively written that defines its function. The specification is ratified as a standard and gradually the existing market conforms to it, leading to new scope for competitive behaviour and innovation. In this model there is no built-in incentive for patent monetisation to be necessary, so standardisation activity is either under mandatory RF terms (example: W3C) or the stakeholders mutually self-select RF terms from multiple options (example: OASIS). Also as a consequence the term "SEP" is largely absent in this model.

OPEN SOURCE & STANDARDS

Open source software became popular in the world of implementation-led standards because its basic assumptions – all rights confirmed in advance of development – were a perfect fit. The fit is so good that skipping the standardisation step and moving directly to a shared implementation now seems more rational for some people. In response, standards bodies are experimenting with hosting implementations, both as a sandbox for standards development and as a reference implementation of ratified standards[12].

The industries using requirements-led approaches have meanwhile found their model is simply not effective for fast-moving, software-developer-led problems that are increasingly arising in parts of their chosen market trajectory[13]. Seeing the open source model's success dealing with these issues, some standards-development groups appear to think they can simply appropriate the open source methodology of the implementation-led world while over-riding that methodology with their pre-existing patent monetisation approach. Such actions fail to understand that the open source methodology fundamentally depends on (and thus guarantees) *prior rights* and does not include or allow for *negotiated rights*.

Modeling that difference as a problem of legal language and precedent has led to impasse. Just coming up with a licensing formula that leaves room for patents has not delivered. The developer-led magic simply won't show up if there is the slightest threat of negotiated bilateral settlements, no matter how the license is phrased. At best we will see more tiny projects under non-reciprocal licenses where all the players are SEP owners or their dependent licensees. None of them will create the hoped-for open source network effects.

WINNING A FIGHT BUT LOSING THE BATTLE

Why is that? There are multiple, inter-linked reasons that, "if you build it, they may not come." In no particular order:

Including FRAND-licensed SEPs in a design guarantees that the patent holders will need private relationships with community participants, creating an uneven playing-field. The terms of those relationships will be private, covered by NDAs. There will be no community transparency either in the relationship or in the behaviour it consequently generates. This undermines trust, since code contributions will

[12] See for example https://oasis-open-projects.org/. OASIS have been a pioneer here for more than a decade, with their multi-IP-mode approach allowing co-existence of standardisers used to both models. They are now moving further, actually hosting implementations under open source terms.

[13] Again, Bessen and Maskin were ahead of us here; "Patents may be desirable to encourage innovation in a static world, but they are less important in a sequential setting, where they may actually inhibit complementary innovation." See Bessen and Maskin, ibid.

be assumed to be made (or withheld) on the basis of the private relationship. Portions of the software could be forced to be proprietary rather than shared due to the inability of the developers to contribute them to the project under sub-licensable terms. Implementors of those portions will need to privately maintain them, increasing their costs. All of these effects would reduce the scope of the innovation and the quality that would otherwise arise from collaboration to magnify and perfect it.

There are thus many reasons not to join such a project. Evidence to date shows very few developers do so unless they have a prior relationship with the patent holders[14].

SOLUTIONS?

As the 5G and IoT marketplaces have emerged as collision points between the two standardisation worlds, the urgency of finding a solution has grown. A legalistic approach will not illuminate the benefits sought. So, what options still exist? In order of increasing probability:

Monetise using an open source model

The software world has always been comfortable with approaches to monetisation that use vehicles other than patents, and today the concept of indirect monetisation is well explored. The most obvious solution for a move into open source is to also use the associated, proven monetisation strategies. The most effective of these all involve making money *with* software rather than *from* software, monetising value it creates rather than directly monetising the software or the ideas it implements. In this world view, patent owners might consider that they are investing their IP to create a market, promote a standard or to catalyse a monetisation strategy, rather than to directly generate (likely smaller) revenues.

Separate concerns

Well over a decade ago, OASIS pioneered the multi-track approach to IP modes within standards. By doing so, they allowed an initially fractious community of stakeholders to converge around the IP modes that

[14] A forthcoming European Commission JRC study provides useful data to substantiate this.

were right for each individual project within the stanwere right for each individual project within the standards body. Perhaps other standards bodies need to explore the same idea?

It's possible for example that some parts of the 5G standards portfolio could be developed under an RF mode (or even a waived rights mode) allowing network effects to be driven through open source sample implementations, while other parts that are tightly bound to physical systems or require very high degrees of conformance for interoperability remain under a FRAND mode.

Don't call it "open source"

Finally, some are bound to choose to ignore the realities of open source and attempt legal frameworks that violate software industry norms. The open source community implores you not to call these experiments "open source"; they are not. Frankly, software projects that retain FRAND licensing can exist, but should not call themselves "open source" as the globally accepted meaning of that term-of-art precludes post-hoc negotiation and patent licensing.

SUMMARY

The issue associated with FRAND and open source is not primarily a legal one; rather, it is a practical matter of standards writers being unable to gain significant value from engagement with the open source community (e.g., for creation of a sample implementation) to generate the expected network effects while also trying to leave room for directly monetising patents. Debating the legal compatibility of open source licenses with royalty-bearing patents in standards is beside the point. One needs to either use a real open source project, or risk that no-one shows up apart from standardisers.

It is possible to use open source licenses on implementations of FRAND-encumbered standards, but one has to choose whether the patents are invested to create direct revenue or a network effect. If you are trying to harmonise patents and open source for direct revenue purposes, you have created a problem with that choice which no amount of careful lawyering will fix.

OPEN SOURCE AND FRAND: WHY LEGAL ISSUES ARE THE WRONG LENS

ofa OpenForum Academy

2019
MARCH