

OSS and FRAND encumbered knowledge production

Executive summary

Gregor Langus and Vilen Lipatov

July 2023

There is a large potential to create value in interaction between open-source software (OSS) and cellular standards development. Some stakeholders in the OSS community, however, claim that this potential is impeded by the incompatibility between OSS licensing and development model and FRAND licensing of Standard Essential Patents (SEPs)—due to legal incompatibility, licensing frictions and cultural reasons. It has thus been proposed that OSS developers should be able to use SEPs without paying a FRAND fee and without negotiations. If this proposal were systemically accepted, it would reduce the ability of SEP holders to monetize their inventions.

Against this background, we tackle two broad questions. First, whether there is indeed a tension—or even incompatibility—between OSS licensing and development model and FRAND-based standardization. Second, whether royalty-bearing FRAND-based licensing in mobile telecommunications could be replaced—without impeding efficiency—with an OSS-like development model and a zero-royalty license for patents essential to the standard.

We start by classifying OSS development and licensing models and setting those against the dominant model of development and licensing of modern cellular standards. We also examine the incentives to contribute to an OSS project. Except in a class of smaller or early-phase OSS projects, we find that commercial motives play an important role in OSS development, although licensing royalties are often not the dominant model of monetization.

We next examine the concerns about potential incompatibility of OSS licensing and development model with FRAND-based standardization. We find these concerns to be exaggerated. Indeed, multiple popular OSS licenses do not explicitly require a patent license and do not seem to imply one; as such, they are compatible with payable FRAND royalties for patents essential to mobile telecommunication standards. Strong evidence of compatibility lies in the multiple successful OSS projects in the field of mobile telecommunications which managed to attract large communities of diverse stakeholders, including those outside the OSS community. In case an OSS project originator is unsure of compatibility of an OSS license with FRAND licensing of patents, they can modify an existing OSS license to explicitly exclude patent grants without stifling software development.

Finally, we identify key factors behind the differences in the incentive to participate in FRAND-based standardization as opposed to participation in an open community OSS project. The factors make it unlikely that zero FRAND and rights-restrictive, OSS-like, licensing and development model would yield good results in the context of mobile telecommunications standardization.

Specifically, first, intrinsic (as opposed to strictly commercial) motives for participation in development can generally be supported better in the OSS context compared to cellular standardization. This is because of the presence of two factors that are necessary for support of intrinsic incentives. Firstly, OSS software developers often have considerable autonomy in relation to task and projects they want to engage on. Secondly, a prompt and precise feedback tends to be available from the community to the developer that establishes a clear locus between successful development effort and recognition. Both autonomy and feedback are more difficult to achieve in the context of cellular standardization.

Second, various larger tasks can effectively be broken into smaller tasks and distributed among disparate developers in the OSS but not in cellular standardization context. As a result, while the advancement of software development may be fast paced, the incremental costs involved in solving an individual task are often relatively low. In turn, the required expected rewards for an individual contributor that induce efficient levels of development effort overall may also be low. In contrast, in some areas of cellular standardization development tasks cannot be easily broken down. In those areas, the incremental costs of advancing a standard may be very high and expected royalty revenues thus play an important role in supporting investment incentives for standard development.

Third, free-riding problem in the cellular standardization context may be more severe than in the OSS development context. The OSS culture, licensing, and the nature of software development allows for an implementation of a quid-pro-quo exchange of making the source code available for use to downstream developers in exchange for future contributions by a downstream user. The compliance with this quid-pro-quo agreement is often relatively easy to verify, as long as the source code, which can be parsed relatively easily, is distributed. Such an agreement would generally be difficult to verify in a cellular standard setting given the multitude of ways in which high-value technical contributions can be implemented. Correspondingly, firms that innovate in the mobile standardization context would have a stronger incentive to free-ride in a zero-royalty licensing regime, while such incentives are muted for OS software developers.

Fourth, delayed disclosure of technical advancements would allow a company integrated into implementation to obtain an advantage in the market, having a working implementation of technology before its rivals. The advancements in the OSS space must be automatically disclosed by the OSS licensing requirement to distribute source code (at least when the license is copyleft) so there is little scope for the developer to delay making available their advancements to others.

Fifth, the knowledge, skill and code created in the context of an OSS project often transfer relatively well across a multitude of software projects, be it open source or proprietary. Software developers can also capitalize their effort on complementary services across a wide array of industries. The mobile technology developers typically do not have a wide set of opportunities, particularly those that are not integrated across the value chain and do not offer a wide variety of end-products.

Finally, the nature of the decision-making process supported by online collaboration platforms in the OSS context often makes reaching consensus required for an incremental advance relatively easier compared to the mobile telecommunications standard setting context. A feature can often be added to software without breaking its backward or forward compatibility. In case of irreconcilable disagreements on the future development directions, a relatively small group of developers can fork the OSS project and continue incremental development as it best sees fit, based on the current version of software. In contrast, “forking” is hardly an option in the standard setting context.