10

Patents and the Defensive Response

Malcolm Bain and P McCoy Smith

10.1 Patents and Software

As discussed in more detail in Chapter 3, the foundation of Open Source licensing is copyright, and in the beginning, consideration of patent rights and patent licences was not paramount. The BSD license,[1] one of the first Open Source licences created (*circa* 1988), states its licence grant as follows:

> Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
>
> . . . Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

[1] There are several different variants of the BSD licence; this text is reproduced from the 'BSD 3-clause license'—the most commonly used BSD variant—as found on the Open Source Initiative's website. <https://opensource.org/licenses/BSD-3-Clause> accessed 12 August 2020.

Malcolm Bain and P McCoy Smith, *Patents and the Defensive Response* In: *Open Source Law, Policy and Practice*. Edited by: Amanda Brock, Oxford University Press. © Malcolm Bain and P McCoy Smith 2022.
DOI: 10.1093/oso/9780198862345.003.0010

> … Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. …

No express mention is made of patents in this grant, although at least one verb—'use'[2]—that is an exclusive right of a patent holder is recited.[3] Similarly, the MIT License, another early Open Source licence created around the same time as the BSD License, states its grant as follows:

> Permission is hereby granted, free of charge, … to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so

Thus, the MIT License[4] uses at least two of the verbs—'use' and 'sell'—that are exclusive rights of a patent holder. At least one commentator has argued that MIT's open ended grant 'to deal in the Software without restriction', followed by exemplary verbs from copyright and patent rights, confers a complete patent licence.[5]

Nevertheless, concerns have long been raised about the scope of patent rights that might be conferred—or might be withheld—in the early Open Source licences. More recent Open Source licences approved by the Open Source Initiative (OSI)—for example the GNU General Public License version 3 (GPLv3 2007) and the Mozilla Public Licence version 2 (MPLv2 2012)—deal quite extensively with patents. For example, relevant portions of the MPLv2[6] read:

> 2.1. Grants
> Each Contributor hereby grants You a world-wide, RF, non-exclusive license:… under Patent Claims of such Contributor to make, use, sell, offer for sale, have

[2] For example, UK Patents Act 1977 § 60; 35 USC § 271(a).

[3] Despite the fact that the general licence grant of the BSD licence is more than thirty years old, there continues to be a debate as to whether any patent rights are conferred by a licensor that chooses to use that licence with their software. Compare David Kappos and Miling Harrington, 'The Truth About OSS-FRAND: By All Indications, Compatible Models in Standards Settings' (2019) 20(2) *Columbia University Science and Technology Law Review* 240–50 with Van Lindberg, 'OSS and FRAND: Complementary Models for Innovation and Development' (2019) 20(2) *Columbia University Science and Technology Law Review* 251–70.

[4] Open Source Initiative, 'MIT License' <https://opensource.org/licenses/MIT> accessed 18 August 2020.

[5] Scott Peterson, 'Why so little love for the patent grant in the MIT License?' *Opensource.com* (23 March 2018) <https://opensource.com/article/18/3/patent-grant-mit-license> accessed 19 March 2021.

[6] Open Source Initiative, 'Mozilla Public License' <http://opensource.org/licenses/MPL-2.0> accessed 12 August 2020.

> made, import, and otherwise transfer either its Contributions or its Contributor Version.
>
> 2.3 Limitations on Grant Scope
> … [N]o patent license is granted by a Contributor … for any code that a Contributor 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.
>
> …
>
> 5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

The GNU General Public License was perhaps the first Open Source License to discuss patent rights in any detail; the second version of the GNU General Public License (version 2, in 1991), indicated that software patents were considered a risk for free software. Version 2 of that license, GPLv2, warned of patent threats in its preamble: '[A]ny free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licences, in effect making the program proprietary. …' GPLv2 includes provisions purporting to deal with patents, in a clause referred to by the Free Software Foundation (FSF)—the GPL's authors—as the 'Liberty or Death clause'. '[T]he clause that says if somebody uses a patent or something else to effectively make a program non-free then it cannot be distributed at all.'[7] '[P]atents not only do not assist in the production of innovative software, they can potentially destroy the free software production system, which is the world's most important source of software innovation.'[8] While over the years Open Source licences themselves have become more sophisticated with regard to patents—at least to the extent that they make explicit that those that contribute code under an Open Source licence do not reserve the right to assert their patents against those making use of their contributions—there is only so much licences can do to guard against the threat of patent assertions, as a licence only binds those that make use of the rights granted under that licence. Although the threat of patent assertions made against Open Source by patent

[7] FSFE, 'Transcript of Richard Stallman at the 2nd international GPLv3 conference; 21st April 2006' <fsfe.org/campaigns/gplv3/fisl-rms-transcript.en.html> accessed 12 August 2020.

[8] Eben Moglen, 'Free software matters: Patently controversial' *Moglen Law* (2001) <http://moglen.law.columbia.edu/publications/lu-16.html> accessed 12 August 2020.

holders who are not participants (via contributions, or via exercising licence grants) has been recognised since at least the release of GPLv2 in 1991, it has only been more recently that initiatives involving the Open Source community have been set up to fend off the threat of the use of patents to limit the creation and use of free software. One example is the Open Invention Network,[9] a patent pool for providing patent non-assertion commitments to the GNU/Linux operating system ecosystem.

What seems paradoxical is that patents and free software appear to share the same basic objective: to promote development and innovation through transparency and disclosure. It is on the basis of disclosing and sharing knowledge (in patent applications) or through access to source code (in Open Source) that new inventions or innovations may be made over existing technology, whether in an incremental manner or by 'intuitive' leaps. Even the legal technique established for promoting inventions via the patent system—that is granting exclusive rights that may be exercised by the inventor to control the exploitation of the invention by others—should not have been a problem: a similar legal framework of exclusive rights in the area of copyright has been used by the free software community from the start as the very basis for granting and ensuring software freedoms.[10]

However, there are significant friction areas between the two models or approaches to innovation; particularly the fact that patents provide for exclusive control over all and any implementations of a patented idea—as that idea is defined in a granted patent claim—and not just an expression of that idea as with copyright, which gives rise to problems and potential legal risks for free software. The purpose of this chapter is to explore these issues, to understand how the Open Source community tries to deal with patents with the aim of ensuring software freedoms, and concludes by commenting on proposals that have been made to remedy the situation and mitigate the risks.

Therefore, in this chapter we first look at why patents are relevant to Open Source—briefly, the question of *software patentability* and the differences with copyright, and then, taking into account the free software development and licensing models, we consider what the impacts are for Open Source: the interrelations and frictions areas between free software licensing models and patents. Next, how patents are dealt with by the community from a structural point of view—particularly patent-related licensing provisions in free software licences—is reviewed. A discussion of the litigation environment, specifically as it relates to patent assertions against Open Source, is discussed. Finally, how the risks posed by patents—or the way patents are wielded—to the Open Source community may be mitigated, if not removed entirely, are summarised.

[9] Open Invention Network <http://www.openinventionnetwork.com> accessed 9 March 2021.

[10] Richard Stallman, 'The Free Software Definition' in *Free Software, Free Society: The Selected Essays of Richard M. Stallman*, 2nd edn (Boston, MA: GNU Press, Free Software Foundation, 2002–10) 43–6, available at <http://www.gnu.org/philosophy/fsfs/rms-essays.pdf> accessed 23 August 2020.

10.2 Patents 101: Why Are Patents Relevant to Open Source?

Patents are exclusionary rights[11] granted to inventors over an invention, conveying to the patent holder rights to exclude anyone else from exploiting the invention as claimed in the patent in the specific territory for which the patent is granted, for a limited period. In return, the patent holder is obligated to provide a full disclosure of the invention to the public. Patents are granted on application to territorial patent offices (e.g., the UK Intellectual Property Office), after examination for patentability, as well as other eligibility criteria, under the applicable rules.

10.2.1 In Europe

Within Europe, patents are regulated on a regional basis by the European Patent Convention (EPC), which creates a European patent with potential effects in the territories of the signatories to the Convention, and on national bases by the corresponding national patent laws, for example the UK Patents Act 1977, or the Spanish *Ley 11/1986 de Patentes*. In this chapter we will mainly comment on the EPC provisions with respect to software, though it is important to note that it is the national courts applying the law of the member states who ultimately decide on patent validity or infringement, though they tend to follow the European Patent Office (EPO) practice and Board of Appeal decisions.

The state of patenting for software has long been controversial, and there are many arguments as to whether software does or should constitute patentable subject matter. Patents are granted for inventions in all fields of technology that are new, involve an inventive step,[12] and are capable of industrial application.[13] The EPC does not define what is an 'invention'. It does, however, provide a negative limitation, giving examples of what are not to be regarded as inventions. Relevant for the purposes of Open Source is the specific exclusion, under Article 52(2)(c) EPC, of 'programs for computers'.

However, this exclusion is then limited by Article 52.3, which provides that these items are excluded 'only to the extent to which a European patent application relates to such subject matter or activities as such'. It is these last two words, 'as such', that have caused an ongoing and acrimonious debate about software patentability under the EPC, and also under the European national legislations, many of which provide a translation or approximation of this double exclusion/limitation

[11] Patents are not 'exclusive' rights, i.e. a positive and exclusive right to do something, but rather a negative right to exclude others from implementing the claims granted in the patent document.

[12] In the US, this concept is referred to as 'non-obviousness'. See 35 USC § 103.

[13] EPC, Article 52. In the US, a related—but not completely analogous—requirement is 'usefulness'. See 35 USC § 101.

with regard to software,[14] and which ultimately is the benchmark against which the validity of the European patent is measured.[15]

It is not the purpose of this chapter to review the situation of software patentability within Europe, as we aim to focus on the interaction between software patents—however well or justifiably granted—and Open Source.[16] Suffice to say that the EPO has long been granting patents over what have been named 'computer-implemented inventions' (CII), on the basis that they are granting patents over inventions that have technical character and a technical effect that goes beyond the normal interaction of the software with the computer, although ironically 'technical' is not defined in the EPC.[17] European national courts (with some reticence, it was once thought, in England and Wales, but that has proven not to be so) are upholding those grants.[18] What is more, in the light of the debate about software patentability, the Enlarged Board of the EPO rejected the EPO President's request to undertake a full review of the situation, at the instigation of the English High Court, considering that the 'case law' created by the EPO Boards of Appeal is sufficiently clear.[19]

> Indeed, if the Boards continue to follow the precepts of T 1173/97 *IBM* it follows that a claim to a computer implemented method or a computer program on a computer-readable storage medium will never fall within the exclusion of claimed subject-matter under Articles 52(2) and (3) EPC, just as a claim to a picture on a cup will also never fall under this exclusion. However, this does not mean that the list of subject-matters in Article 52(2) EPC (including in particular 'programs for computers') has no effect on such claims. An elaborate system for

[14] For example, Spanish Patent Act 11/1986, art 4.

[15] The proposed Unified Patent Court, approved by the European Council of Ministers and European Parliament, does not exclude software patents per se, but does have limits to enforcing such patents consistent with European Parliament directives allowing for reverse compilation and interoperability. Agreement on a Unified Patent Court, UPC/en 34 n. 1 (19 February 2013).

[16] There are a significant number of thoughtful papers written on this subject. See Noam Shemtov, 'Software Patents and Open Source Models in Europe: Does the FOSS Community Need to Worry About Current Attitudes at the EPO?' (2010) 2(2) *Journal of Open Learning, Technology & Society (JOLTS)* 151–64; Avi Freeman, 'Patentable Subject Matter: The View From Europe' (2011) 3(1) *Journal of Open Learning, Technology & Society* 59–80; Colleen Chien, 'From Arms Race to Marketplace: The Complex Patent Ecosystem and Its Implications for the Patent System' (2010) 62 *Hastings Law Journal* 297–356; Mark Lemley, 'Software Patents and the Return of Functional Claiming' 2013 *Wisconsin Law Review* 905–64, available at <http://ssrn.com/abstract=2117302 or <http://dx.doi.org/10.2139/ssrn.2117302> accessed 21 July 2022.

[17] EPO Board of Appeal Decisions: Computer program I/IBM (T1173/97) and Computer program II/IBM (T 0935/97). See EPO, Guidelines for Examination in the European Patent Office, G-II 3.6 (2019), available at <https://www.epo.org/law-practice/legal-texts/html/guidelines/e/g_ii_3_6.htm> accessed 24 August 2020.

[18] For example, *Aerotel Ltd v Telco Holdings Ltd* [2007] RPC 7; *Macrossan's Application* 2006 [EWCA], followed by *Symbian Ltd v Comptroller General of Patents* [2008] EWCA Civ 1066; *Halliburton Energy Inc's Patent* [2011] EWHC 2508 (Pat).

[19] Enlarged Board of Appeal Opinion G3/08. For commentary, see Freeman, 'Patentable Subject Matter: The View From Europe', note 16.

> taking that effect into account in the assessment of whether there is an inventive step has been developed, as laid out in T 154/04, *Duns*. While it is not the task of the Enlarged Board in this Opinion to judge whether this system is correct, since none of the questions put relate directly to its use, it is evident from its frequent use in decisions of the Boards of Appeal that the list of 'non-inventions' in Article 52(2) EPC can play a very important role in determining whether claimed subject-matter is inventive … It would appear that the case law, as summarised in T 154/04, has created a practicable system for delimiting the innovations for which a patent may be granted.

In practice, as stated on various occasions by examiners of the EPO,[20] while they consider software-based inventions with technical effect as patentable subject matter, many software patent applications are being rejected on the basis of lack of novelty (the second hurdle, considering 'patentable subject matter' as the first) or lack of inventive step (the third hurdle).[21] In particular, mere computer- or software-based automation of constraints imposed by non-technical aspects—specifically those that are excluded by the EPC—notably mental acts, games, business methods, or methods for presenting information, are allegedly not being granted patent protection.[22]

10.2.2 In the US

In the US, for many years the leading decisions in the debate on software patentability were the US Supreme Court's decision in *Diamond v Diehr*[23] and subsequently *State Street Bank & Trust v Signature Financial Services*[24] where the Court of Appeals for the Federal Circuit held that a computerised algorithm for managing an investment fund structure constituted patentable subject matter which should be evaluated under the usual US tests of usefulness, novelty, and non-obviousness.[25] Subsequently, in *In re Bilski*, the Federal Circuit seemed to have

[20] See, e.g., EPO presentation by Eugenio Archontopoulos, 'Spot the Differences, A Computer-implemented Invention or a Software Patent?' (6th Annual Conference of the EPIP Association, Brussels, 2011) <https://www.researchgate.net/publication/230818897_Spot_the_difference_a_computer-implemented_invention_or_a_software_patent> accessed 16 June 2022.;

[21] In particular, features making no contribution to the technical character cannot support the presence of inventive step (*Comvik* (T0641/00) and *Duns Licensing* (T0154/04)). Also, Hanon 'What makes an Invention—How patent applications are examined at the European Patent Office', see note 20, and Archontopoulos, 'Spot the Differences, A Computer-implemented Invention or a Software Patent?' see note 20.

[22] *Ricoh Decision* T 03/0172; *Hitachi Decision* T 03/0258.

[23] 450 US 175 (1981).

[24] 149 F.3d 1368 (Fed Cir 1998) cert denied; 525 U.S. 1093 (1999).

[25] See Christopher Ogden, 'Patentability of Algorithms after State Street Bank: The Death of the Physicality Requirement' (2000) 10(82) *Journal of Patent and Trademark Office Society* 491–513.

begun to apply a more strict approach towards software patentability:[26] it found that a patent on a method of hedging financial risk in commodity trading claimed 'neither a new machine nor a transformation of matter', and thus was too abstract and non-patentable subject matter. However, the US Supreme Court then mitigated this analysis, to a certain extent, holding that the 'machine-or-transformation test' is not the only test for determining the patent eligibility of a process (but rather 'a useful and important clue ... an investigative tool, for determining whether some claimed inventions are processes under §101').[27] And in *Mayo Collaborative Services v Prometheus Laboratories, Inc*,[28] the US Supreme Court reaffirmed the judicially created exception that makes 'laws of nature, natural phenomena, and abstract ideas' ineligible for patenting, leading some to believe that there was an opening of the judicial 'door' to making the argument that software code is merely a series of mathematical algorithms and, as such, a description of abstract laws of nature.

The US Supreme Court's later decision in *CLS Bank v Alice Corp*[29] buttressed the importance of the non-software decision in *Mayo,* on software-related patentability determinations. Much like *Bilski*, *Alice* related to implementation of a business method: in *Alice*, a software-implemented system for managing escrow debt. In finding that particular invention patent-ineligible, the US Supreme Court stated that a two-step '*Mayo* framework' should be used in evaluating patent eligibility questions: the first step is to determine whether the challenged patent claim contains an 'abstract idea', such as an algorithm, method of computation, or other general principle; if it does, then the second step is to determine whether the challenged patent adds to the abstract idea an 'additional feature' that embodies an 'inventive concept'. [30] If so, the challenged claim is patent-eligible.[31]

[26] *In re Bilski* 545 F.3d 943 (Fed Cir 2008) (en banc). For comment, see, e.g., Dennis Crouch, 'In re Bilski: Patentable Process Must Either (1) Be Tied to a particular machine or (2) Transform a Particular Article' *PatentlyO* (30 October 2008) <http://www.patentlyo.com/patent/2008/10/in-re-bilski.html>> accessed 19 March 2021.

[27] *Bilski v Kappos*, No 08-964, 561 U.S. (2010). Comment by Crouch, 'In re Bilski', see note 26.

[28] *Mayo Collaborative Services v Prometheus Laboratories*, *Inc* 566 US (2012). Decision available at <http://www.supremecourt.gov/opinions/11pdf/10-1150.pdf> accessed 19 March 2021.

[29] 573 US 208 (2014).

[30] The addition of the 'inventive concept' test to patent eligibility determinations under *Alice* has been widely criticised as improperly conflating the non-obviousness requirement of 35 USC § 103 with the general patent eligibility requirements of 35 USC § 101. *See* Paxton Lewis, 'The Conflation of Patent Eligibility and Obviousness: Alice's Substitution of Section 103' (2017) 1 *Utah OnLaw: The Utah Law Review Supplement* Article 1, 13-32..

[31] The *Bilski-Mayo-Alice* triumvirate of US Supreme Court eligibility cases may not have entirely settled the question of how to evaluate whether a patent is directed to merely an 'abstract idea' and thus patent-ineligible. The Court of Appeals for the Federal Circuit's decision in *American Axle & Manufacturing, Inc. v Neapco Holdings LLC*, 939 F.3d 1355 (Fed. Cir. 2019) has been argued to import yet another statutory requirement—enablement under 35 USC § 112—into the 'abstract idea' analysis. David Taylor, 'Opinion Summary—American Axle & Manufacturing, Inc. v. Neapco Holdings LLC' *Federal Circuit Blog* (31 July 2020) <https://fedcircuitblog.com/2020/07/31/opin

Despite continued questions about the manner in which to evaluate the eligibility for patenting of software in the US, the number of 'software patents' being granted does not appear to have slowed down. This has also led to questions not only about whether many of the 'software patents' granted in the US—particularly those in the period between the *State Street* and *Bilski* & *Alice* decisions—are weak, if not trivial, and might ultimately fail upon a challenge as to eligibility under the current, or to be outlined in the future, test. In the interim, commentators have remarked upon the creation of patent 'thickets' of overlapping and poor-quality patents, which close down innovation and may make it difficult to operate in the software sector.[32]

So, all in all, current industry practice, the pressure from large software industry companies and other non-industry players such as non-practising entities, combined with the lack of resources and time for reviewing patents at the patent offices and the lack of access to relevant prior art in the field,[33] together mean that software patents have been and are still being granted over software implemented processes and methods on both sides of the Atlantic as well as in Japan, another key jurisdiction. Specific examples include security algorithms for encryption, audiovisual data codification and decodification ('codecs'), online data back-up, graphical user interface features, 'one-click' online shopping systems, frames for displaying information on computer interfaces, and the list goes on.[34]

ion-summary-american-axle-manufacturing-inc-v-neapco-holdings-llc/> accessed 28 August 2020. There seems to be some likelihood that the contours of the test for determining patent eligibility for claims argued to be directed to 'abstract ideas' have yet to be fully defined in the US, and there was thought to be a reasonably likelihood that the US Supreme Court might take up the *American Axle* case to further clarify patent eligibility—which might include clarifying patent eligibility for software in the US. Eileen McDermott, 'Solicitor General Tells SCOTUS CAFC Got it Wrong in American Axle, Recommends Granting' *IP Watchdog* (24 May 2022) https://www.ipwatchdog.com/2022/05/24/solicitor-general-tells-scotus-cafc-got-wrong-american-axle-recommends-granting/id=149248/> accessed 14 June 2022 (noting that the Solicitor General of the US—the office which offers the US Government's position on cases before the Supreme Court of the US—had requested that that court reexamine patent eligibility through that case). Much to the surprise of many who felt that the *American Axle* case was an ideal vehicle for further clarifying (or possibly changing) the patent-eligibility standards in the US, the US Supreme Court ultimately declined to review that decision. See Blake Brittain, 'U.S. Supreme Court rejects American Axle case on patent eligibility', *Yahoo! News* (30 June 2022) <https://news.yahoo.com/u-supreme-court-rejects-american-171958332.html> accessed 30 June 2022.

[32] Rosa Ballardini, 'The Software Patent Thicket: A Matter of Disclosure' (2009) 6(2) *SCRIPTed* <https://script-ed.org/wp-content/uploads/2016/07/6-2-Ballardini.pdf> accessed 19 March 2021, DOI: 10.2966/scrip.060209.207.

[33] Software patenting has a long history, dating back to at least the late 1960s. Gene Quinn, 'The History of Software Patents in the US' *IP Watchdog* (30 November 2014) <https://www.ipwatchdog.com/2014/11/30/the-history-of-software-patents-in-the-united-states/> accessed 19 March 2021. Nevertheless, for quite some period, there was little 'prior art' previously published in a meaningful manner —particularly in patent office databases—for disclosure against subsequent patenting.

[34] An interesting series of software patents can be found at the End Soft Patents wiki, 'Example software patents' <http://en.swpat.org/wiki/Example_software_patents> accessed 19 March 2021.

10.2.3 Differences with copyright

When a patent is granted on a software-based invention or CII, it doesn't just grant exclusionary rights over the exploitation of a specific implementation of that invention, but *any* implementation of the invention that meets all the elements of any claim in that patent—it protects the functional features of the 'invention', the underlying methodologies, in any manner or form of expression. This is in contrast with copyright protection, which only protects the expression embodied in either the source or binary code of the software.

This means that while copyright protection is generally weaker than patent protection, it is more specific, referring only to the concrete expression of the code developed by the programer. This has the advantages of providing legal certainty with regard to what exactly is prohibited or restricted by copyright, particularly verbatim copying,[35] and what is permitted—alternative or clean room development of similar functions, incremental development of additional functionalities, or complementary development of other programs using software interfaces and interoperability characteristics. Being more specific and restricted to expression, copyright enables a much broader range of alternative implementations and improvements of a same idea or function, through different algorithms, coding languages, or architectures.

There is a crucial distinction between the way patent and copyright concepts respond to the challenge free software poses. Copyright law is primarily intended to cover expressions, not ideas. So, if in a particular instance software copyright inhibits progress in making better, more reliable, or more effective software, the inhibition can be overcome: it is always possible for programers, with sufficient guidance and appropriate measures to prevent copying, to sit down and rewrite from scratch whatever program needs to be available in a freely modifiable version. This may be time-consuming, but it cannot be forbidden. Patent law, in contrast, prohibits anyone from practising the claimed subject matter of the patent without licence. It does not matter how you came by the idea the patent discloses, even if you invented it for yourself in complete ignorance of the patent and any prior art it references: without a licence you cannot implement, in *any* way, the claimed subject matter of what may be quite general claims.[36] This enables patent holders potentially to restrict competition by other developers wishing to implement similar functionalities in their own programs using completely different code expressions. Patents can also seem vaguer or less definite, particularly in the way software patents have been drafted in the time before the *Bilski*, *Mayo*, and *Alice* decisions in

[35] To a major extent, although there are always questions about non-verbatim copying and derivative works which the courts deal with on a fairly regular basis.

[36] Eben Moglen, 'Free Software Matters: The Patent Problem' *Moglen Law* (9 October 2000) <http://moglen.law.columbia.edu/publications/lu-05.html> accessed 19 March 2021.

the US. It is often quite difficult to determine exactly if the implementation of a software process may infringe an existing patent, as there is no way to 'clean room' develop code to avoid a patent. This creates significant legal uncertainty.

The law on software patents, unlike software copyright in jurisdictions like the European Union (EU), provides no exemption for interfaces. As an interface is a set of definitions or specification of a method or process (for using the program or data), it is particularly prone to being 'patentable'. So not only is there the potential for patents foreclosing specific computer-based processes but also there may be patents over software interfaces that may be required to connect with and use software processes.

Another significant difference between copyright and patents (relevant for Open Source) is the characteristics and structures of creation and ownership of rights: copyright in a software program belongs originally to its creator (or the company where the creator works), who has invested time and effort in developing the code, and the rights may be licensed or assigned, usually to someone who wishes to use or further develop the program. Thus, copyright rights are generally held by parties interested in exploiting the software. A patent is first owned by its inventor, who may or may not be a software developer. As there is not necessarily any 'software development' involved in inventing a process that may be embodied by software, the patent rights may be held by any party, who may or may not be interested in implementing the patented process or method, and in some cases may be held by a party interested in controlling or precluding the use by others of the patented process or method.

This situation is illustrated by what have been now called 'non-practising entities' (NPEs) (often pejoratively described as 'patent trolls').[37] These are persons or companies that do not have any particular interest themselves in exploiting the software that implements the patented processes, but only in asserting the patent rights against participants in the software industry interested in the invention, as a mechanism to extract royalty or other payments. NPEs also are less susceptible to external pressures that would otherwise forestall their use of patents to inhibit software use and deployment—because they have no business other than to assert patents, counter-assertions or business pressures are generally ineffective. While assertions of this sort are a legitimate function of patent rights, this creates a significant imbalance in the software sector and can constitute a major block on innovation.[38] This is not to say that there are not 'copyright trolls', monetising copyrights

[37] Wikipedia, 'Patent troll' <http://en.wikipedia.org/wiki/Patent_troll> accessed 19 March 2021.

[38] For commentary, see James Bessen, Michael Meurer, and Jennifer Ford, 'The Private and Social Costs of Patent Trolls' (19 September 2011) Boston University School of Law, Law and Economics Research Paper No. 11-45 <http://ssrn.com/abstract=1930272> or <http://dx.doi.org/10.2139/ssrn.1930272>.

through litigation.[39] We will comment further on this later, when looking at the interactions and frictions between Open Source and patents.

10.2.4 Patent remedies

The remedies available to patent holders in the case of infringement are important to understand the potential effect of patents against Open Source. National courts in Europe are competent to hear infringement cases and determine remedies of both the national equivalents of European patents and patents issued directly by their national offices. However, except for very limited circumstances, the national court's decision will only apply in its territory, and if the infringement occurs in several states, then proceedings would have to be brought independently in each country.[40] This is likely to change when the Unified Patent Court (UPC)[41] comes to fruition. The UPC will be a specialised patent court with exclusive jurisdiction for litigation relating to European patents and European patents with unitary effect (unitary patents). In practice, absent a UPC, Germany seems to be one of the favourite states to start infringement proceedings, as those proceedings are relatively cheaper and faster there (many decisions are made under the fast injunction

[39] In the early 2000s, SCO was accused of being a 'copyright troll' against UNIX and Linux. David Kravets, 'Copyright troll loses high-stakes Unix battle' *Wired* (31 March 2010) <https://www.wired.com/2010/03/unix-copyrights/#ixzz0yUsnFxzG> accessed 28 August 2020. More recently, an individual named Patrick McHardy has been accused of being a 'copyright troll' as the result of GPL violation lawsuits filed in Germany. Ieva Giedrimaite, 'Copyright trolling: Abusive litigation based on a GPL compliance' *The IP Kitten* (24 February 2019) <https://ipkitten.blogspot.com/2019/02/copyright-trolling-abusive-litigation.html> accessed 28 August 2020.

[40] It is possible to bring action against the defendant in its jurisdiction of residence and the local courts may in this case handle infringements across the relevant EU territories based on the origin of infringement with the defendant in its residential jurisdiction. There is also a practice in Dutch courts of granting cross-border injunctions in patent cases, although the circumstances under which can be done are likely limited to summary proceedings. Renaud Dupont, 'Cross-border injunctions are back in the Netherlands' *Lexology* (27 September 2011) <https://www.lexology.com/library/detail.aspx?g=2b5e8ef1-bf5a-46fd-8499-61f766c83424> accessed 29 August 2020. See also *Solvay SA v Honeywell Fluorine Products Europe BV*, Case C-616/10 (ECJ 12 July 2012).

[41] The Agreement on the UPC was endorsed by EU ministers in the Competitiveness Council on 10 December 2012 and by the European Parliament on 11 December 2012; because of Brexit and an adverse ruling from the German Federal Constitutional Court, the Unified Patent Court was for some time believed not to have achieve sufficient ratification to commence, and many predicted that it would not be instituted. James Nurton, 'German decision puts Unified Patent Court agreement in jeopardy' *IP Watchdog* (20 March 2020) <https://www.ipwatchdog.com/2020/03/20/german-decision-puts-unified-patent-court-agreement-jeopardy/id=120013/> accessed 29 August 2020. However, Germany eventually ratified the UPC, setting the UPC up to commence operation in 2022 or 2023—although there still remain questions as to whether the UK is required to ratify and participate in the UPC. Christoph Crützen, Benjamin Beck, and Maximilian Kücking, 'Germany Ratifies EU Unified Patent Court (UPC) Agreement, but Prospects for the UPC Remain Uncertain', *Mayer Brown blog* (18 August 2021) <https://www.mayerbrown.com/en/perspectives-events/publications/2021/08/ger-germany-ratifies-eu-unified-patent-court-agreement> accessed 14 June 2022.

procedure), something that has been seen in the case of the *Apple v Samsung* proceedings relating to Samsung's 'Galaxy' tablet.[42]

Remedies have been broadly harmonised across the EU through Directive 2004/48/EC of the European Parliament and of the Council of 29 April 2004 on the enforcement of intellectual property (IP) rights.[43] Remedies include both precautionary measures, such as preliminary injunctions and seizure, as well as permanent orders and monetary damages.

As the patent holder's main goal is to stop the infringing party's actions, it will mainly aim for preliminary and then permanent injunctions to cease the manufacture, distribution, commercialisation, and use of the infringing product. In addition, at the preliminary stage the patentee may request an order to seize or produce for audit products, tools (including computer equipment), production plants, books of account, invoices, and advertising materials, the latter in order to collect documentary evidence of the infringement and its extent; and a blocking order to stop imports at the national borders. In the extreme, a patentee may also request freezing the allegedly infringing party's bank accounts. Thereafter, when infringement is finally determined, the rights holder can request a declaration of the validity of the patent and the destruction of the infringing items.

If infringement is found, damages may be applied for to compensate for the infringing activities, either as accounts for profits made, monetary compensation for lost profit of the patent holder, or the fees the patentee would have charged for granting a licence (probably the preferred method, as proving lost profits or trying to work out the infringer's illegitimate profit made on the basis of the patented item, can be difficult).

We will see in the following section how difficult it is to apply these concepts in the Open Source software context. Not only is identifying infringers of a CII implemented in Open Source potentially unknown or difficult to identify or locate (assuming that the Open Source project is the 'person' infringing a third party's patent), but also it can be extraordinarily difficult to prevent distribution of intangible goods (that may infringe on patent rights) on the Internet.[44]

This is not the case when the software is embedded in hardware devices, such as smart phones, set-top boxes, or routers, where the patent holder may pursue any member of the supply chain (in particular the retailer and the importer) to obtain the injunctive relief and subsequent claim for damages. This is probably one of the

[42] See Chris Foresman, 'Apple stops Samsung, wins EU-wide injunction against Galaxy' *ArsTecnica* (9 August 2011) <http://arstechnica.com/apple/2011/08/samsung-facing-eu-wide-injunction-against-galaxy-tab-101> accessed 14 June 2022. Germany is a preferred venue, see comment by Kevin O'Brien, 'German Courts at Epicenter of Global Patent Battles Among Tech Rivals' *New York Times* (8 April 2012) available at <http://www.nytimes.com/2012/04/09/technology/09iht-patent09.html> accessed 19 March 2021.

[43] *Official Journal of the European Union* L157 of 30 April 2004.

[44] See, e.g., how OpenSuSE community deals with audiovisual codecs encumbered by patents: OpenSuSE wiki, 'Restricted formats' <https://en.opensuse.org/Restricted_formats> accessed 19 March 2021.

reasons patent litigation has been so popular in the mobile device industry, as there are specific goods or devices to identify for remedial action.

Thus, there are a series of reasons why patents are relevant to software, in particular their very existence with respect to software implemented inventions, their nature and scope, and their differences with copyright, many of which, as we will see next, enter into conflict with the principles and reality of Open Source.

10.3 Patents and Open Source Interactions

To understand the interactions between Open Source software and patents, we must briefly review the nature and characteristics of Open Source and its development process. As we will then see in this section, these are not particularly well-suited to the patent system (as legislated and practised), leading to a variety of areas and types of friction. In the next section of this chapter, we will look at how the Open Source community tries to deal with these frictions, both in the licensing regimes and in practice.

10.3.1 Development and Innovation in Open Source

Open Source is software that is distributed under an Open Source licence. These licences are broad, RF licences that allow all persons to use, copy, modify, and distribute the original code and its derivative works.[45] Thus Open Source is characterised by the granting to others of the ability to exploit the software, with access to its source code as a requirement to be able to enjoy those rights.

Any Open Source licence is in fact a practical expression of the ideals and objectives of the software creators, using copyright rights (and in some cases, patent rights) to allow and enforce openness and freedom with respect to the software code and the knowledge contained therein. Open Source licensing increases public accessibility to this knowledge. Under copyleft licences,[46] a sub-group of Open Source licences, this knowledge and these freedoms to exploit and innovate are guaranteed for all third parties through obligations to maintain the free software licensing terms in downstream distributions of the product and its derivative works.

In practice, this usually leads to a decentralised software development model, the 'bazaar', as Eric Raymond has called it,[47] whereby developers from all parts of the world may participate in and contribute to an Open Source project. These

[45] Stallman, 'The Free Software Definition', note 10. See also Open Source Initiative, 'Open Source Definition' available at <https://opensource.org/osd> accessed 19 March 2021.

[46] Stallman, 'The Free Software Definition', note 10.

[47] Eric Raymond, 'The Cathedral and the Bazaar' (2000) <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar> accessed 19 March 2021.

participants form what has generically been called the 'community' of the project, and these communities together form the 'Open Source community' or movement as a whole. These communities are extremely heterogeneous, including individual programers and users, institutions, companies, and public bodies, and can be formed by one or two persons, or a significant number of participants such as the Open Document or GNU/Linux communities.[48] The community participants, acting usually remotely over the web, maintain, develop, and correct the project software according to a roadmap that may or may not be an agreed 'master' document. In some communities, such as the Mozilla, Ubuntu, or Alfresco projects, the project may be led or structured by a foundation or corporate entity, which guides development and may exploit the software (or services based on the software) commercially.

Innovation in these communities is varied, either incremental—developers building on previous contributions made by themselves or other participants, or complementary—developing new functionalities and modules through standard and open interfaces. However, in all circumstances, innovation is based on the principles of freedom and openness: taking advantage of broad rights to copy, share, and improve the code, along with open access to the source and interoperability information of the project code.[49]

The certainty provided by the standardised copyright licensing terms established by the project Open Source licence provides reliability and trust among the participants, increasing network effects and providing a strong basis for further innovation.[50] In transaction cost analysis terms, this 'lowers the informational and transactional cost of licensing, as the terms are standard and transparent to all parties, so there is no information asymmetry and no need to negotiate terms'. [51]

10.3.2 Frictions with the patent regime: differences in concept

This form of innovation through sharing, however, runs counter to the justification for patent protection, which is based on the historical and theoretical foundation

[48] See, e.g., Linux Foundation Annual Report 2020, estimating 890,000 contributors, including 44,000 'core developers'. Linux Foundation, 'Annual Report 2020' (2020) <https://www.linuxfoundation.org/wp-content/uploads/2020-Linux-Foundation-Annual-Report_120520.pdf> accessed 19 March 2021.

[49] Chris diBona, 'Introduction' in Chris DiBona, Sam Ockman, and Mark Stone (eds), *Open Sources: Voices from the Open Source Revolution* (Sebastopol, CA: O'Reilly Media, 1999) 1–18.

[50] Notwithstanding the difficulties of interpreting certain licences in certain conditions, for example, the copyleft scope of the GPL. However, the most vibrant Open Source community, the Linux Community, uses the GPLv2 as its legal foundation, showing that this is not an impediment to innovation and sharing.

[51] Jason Schultz and Jennifer Urban, 'Protecting Open Innovation: The Defensive Patent License as a New Approach to Patent Threats, Transaction Costs, and Tactical Disarmament' (2012) 26 *Harvard Journal of Law and Technology* 1, 15.

of IP rights regimes, that of providing economic incentives to creativity and innovation through the artificial creation of exclusivity,[52] although this exclusivity does eventually end and the patent subject matter enters the public domain, upon expiration of a patent's term. Yochai Benkler, among others, has clearly argued that in the information society, as exemplified by free software production models, this justification is not necessarily correct, as there are (many) other incentives to innovation, including curiosity, need, benefits to reputation, the simple desire to share knowledge, or stimulating demand for a related product or service.[53]

Patents also offer the risk of over protection: going back to the historical debate of how to protect and incentivise the creation of software, there were arguments against the broad protection granted by patent rights over 'any' implementation of a particular process, its functionalities, its interoperability, and the impossibility of carrying out reverse engineering, as being too wide and hindering competition and innovation in this sector.[54] Recognising this, the copyright legal regime for software—at least in the EU—provides express exclusions for interoperability and reverse engineering to study the principles and ideas behind a software program, for example to be able to reproduce in a new manner its functionalities.[55]

This is particularly important for Open Source, one of whose main areas of development is the reverse engineering of proprietary software formats and functionalities, to create and distribute under Open Source licence terms both programs with similar features and software that is interoperable with proprietary formats (e.g., OpenOffice.org/LibreOffice or SAMBA).[56]

[52] See, e.g., Paul David 'Intellectual Property Institutions and the Panda's Thumb: Patents, Copyright, and Trade Secrets in Economic Theory and History' in Mitchel Wallerstein, Mary Mogee, and Robin Schoen (eds), *Global Dimensions of Intellectual Property Rights in Science and Technology* (National Academy Press: Washington, DC, 1993) 19–62; or Gillian Hadfield, 'The Economics of Copyright' (Columbia University Press: New York, 1992) 38 *Copyright Law Symposium* 1-46; reviewed in Christian Handke, 'The Economics of Copyright and Digitisation: A Report on the Literature and the Need for Further Research' (London: World Economic Press, 2010). For counter arguments, see Michele Boldrin and David Levine, *Against Intellectual Monopoly* (Cambridge: Cambridge University Press, 2008) esp. ch 7, 'Defenses of Intellectual Monopoly'.

[53] Yochai Benkler, *The Wealth of Networks: How Social Production Transforms Markets and Freedom* (New Haven, CT: Yale Press, 2006) at 63. Collaborative development models are also described in various articles in DiBona et al. (eds), *Open Sources: Voices from the Open Source Revolution,* see note 49; and, e.g., Chris DiBona, 'Open Source and Proprietary Software Development' in Chris DiBona, Danese Cooper, and Mark Stone (eds), *Open Sources 2.0: The Continuing Evolution* (Sebastopol, CA: O'Reilly Media, 2006) 21–36.

[54] See debates of WIPO, Advisory Group of Governmental Experts on the Protection of Computer Programs, *Copyright* (WIPO's monthly bulletin) March 1971, 5–40; and WIPO Group of Experts on the Legal Protection of Computer Software, *Draft Treaty for the Protection of Computer Software* (Geneva, 13–17 June 1983).

[55] See WIPO Model Provisions for the Protection of Software 1983 and, e.g., EC Software Directive, arts. 5 and 6.

[56] Libre Office: <http://www.libreoffice.org> and Samba: <http://www.samba.org> accessed 19 March 2021.

In *SAS Institute v Worldwide Programming*,[57] the European Court of Justice (ECJ) reviewed the question of the protection by copyright of software functionalities, in the context of innovation and technical progress, concluding that:

> [o]n the basis of those considerations, it must be stated that, with regard to the elements of a computer program which are the subject of Questions 1 to 5, neither the functionality of a computer program nor the programming language and the format of data files used in a computer program in order to exploit certain of its functions constitute a form of expression of that program for the purposes of Article 1(2) of Directive 91/250.
>
> As the Advocate General states in point 57 of his Opinion, to accept that the functionality of a computer program can be protected by copyright would amount to making it possible to monopolise ideas, to the detriment of technological progress and industrial development.[58]

However, what is granted by the copyright regime (reverse engineering and interoperability), can be taken away by the patent regime. And although the copyright and patent regimes should ideally be complementary and non-exclusionary, an outcome in which one regime gives a right that the other regime takes away seems illogical taking into account that the objectives of the two regimes, to incentivise and reward creativity and innovation, are basically the same.

10.3.3 Patent frictions in practice

Not just on a theoretical basis but also in practice, there are a significant number of friction areas between the legal regime for patents, and Open Source and its production and distribution models.

First, as regards obtaining patents—if the Open Source community did ever want to patent inventive processes of a project—in environments where innovation is incremental and distributed throughout a community, it may be difficult if not impossible to determine who would qualify as an inventor. And who ultimately should be the beneficiary and rights holder of the patent rights resulting from community development? There is often no such figure or entity to hold them, other than all the individuals who contributed to the conception of the invention itself.[59]

[57] *SAS Institute Inc v World Programming Ltd*, C-406/10.

[58] ECJ decision C-406/10, paras. 39, 40.

[59] Joint ownership of a patent by a collection of developers can introduce complexities (or simplicities), depending on the jurisdiction in which the patent is granted. For example, in the US, all named inventors would have the right to exploit (use for their own purposes) the patent, including licensing it to others—including under an Open Source licence—without having to account (i.e. pay) any of the other inventors. This is not the case in other countries, including the UK (where consent is required from other inventors for an inventor to grant licences). See Raymond Millien, 'The Default Law of Joint

Second, from a risk analysis point of view, the risk of infringing copyright in software is far lower than the risk of infringing a patent. Copyright infringement can be avoided by implementing good development practices and (if need be) creating new and independent versions of copyrighted software. With regard to Open Source licensed code, it is in fact quite difficult to infringe copyright, as most exclusive copyright rights in the original code that you may be working on or with, are granted. Conversely, a patent over a software process can stop anyone from making, using, or selling the patented invention, even if there is no copying of the inventor's original software (if any). This means that it may be impossible to avoid infringing a patent regardless of how much care is taken, particularly essential patents on standards. In the early 2000s, there was at least one published assertion that the GNU/Linux operating system might infringe some 280 software patents,[60] although there was substantial debate about the meaning of that assertion.[61] What's more, the source code availability of Open Source allows a patent-based plaintiff to evaluate infringement easily, while a reverse-engineered patent infringement evaluation of binary code would be more difficult. 'Software patents are dangerous to software developers because they impose monopolies on software ideas. It is not feasible or safe to develop nontrivial software if you must thread a maze of patents.'[62] Moreover, it is argued that this situation is worse for Open Source than for proprietary projects.[63] As we have commented, Open Source is often developed by many people—volunteers—in 'open' communities. These communities rarely have any company or institution providing (legal or financial) support, and thus the individual developers might be more vulnerable to litigation. They certainly don't have the financial resources to cover the cost of dealing with patent issues, which can cost thousands if not millions of Euros. However, a counter-argument is that these individuals are not worth pursuing by patent holders, which may be one of the reasons that to date there are few if any patent-based cases against non-commercial Open Source projects.[64]

However, the counter to this is that any corporate end-users could be viewed as vulnerable to attack. While copyright focuses on the potentially infringing

IP Ownership' *IP Watchdog* (18 February 2016) <https://www.ipwatchdog.com/2016/02/18/the-default-law-of-joint-ip-ownership/id=66154/> accessed 19 March 2021; UK Patents Act 1977 (as amended) § 36-2(a) (1 October 2014).

[60] See Daniel Lyons, 'Linux Scare Tactics' *Forbes Magazine* (8 February 2004) <http://www.forbes.com/2004/08/02/cz_dl_0802linux.html> accessed 19 March 2021; and *Open Source Risk Management Position Paper—Mitigating Patent Risks* (2 August 2004).

[61] Steven Vaughn-Nichols, 'Author of Linux Patent Study Says Ballmer Got It Wrong' *EWeek* (19 November 2004) <https://www.eweek.com/servers/author-of-linux-patent-study-says-ballmer-got-it-wrong> accessed 29 August 2020.

[62] Richard Stallman, 'Europe's 'Unitary Patent' Could Mean Unlimited Software Patents' <http://www.gnu.org/philosophy/europes-unitary-patent.html> accessed 19 March 2021.

[63] Jason Morgan, 'Chaining Open Source Software: The Case Against Software Patents' (1999) <https://groups.csail.mit.edu/mac/projects/lpf/Patents/chaining-oss.html> accessed 19 March 2021.

[64] For more detail about patent litigation against Open Source, see section 10.6 later in this chapter.

copying, transformation, and distribution of software (thus acts carried out by persons in the software industry), any person who also *uses* software that infringes a patent is liable and can have monetary damages and an injunction awarded against them, regardless of whether they were aware of the patent or had any intent to infringe it, and regardless of whether they have any technical or other expertise in dealing with patents. This has a significant impact across industry, raising development expenses, and increasing legal risks and insurance premiums. This also hinders the uptake of the Open Source projects' output through fear of litigation, or making it more expensive by encouraging participants to take a royalty-bearing patent licence.

For a non-commercial Open Source project (and most commercial ones too), taking a patent licence can introduce difficulties. Patent licences and associated royalties are usually based on usage, and an Open Source project rarely if ever knows how its software is used, improved, or redistributed. In addition, in the event of using any Open Source under copyleft licences, in particular GPLv2, the patent licence would have to contemplate redistribution of the code unencumbered by any downstream patent restrictions so to enable the code to remain free; the patent holder would have to be willing to grant wide downstream user rights, something they are unlikely to be willing to do, absent any numerical data on usage.[65]

> We cannot just buy a patent license, because though free software isn't always free like free beer, it cannot exist at all unless it is free like free speech: everyone has to be allowed to take free code from one place and use it in another, or build on it, so long as she is willing to share and share alike.[66]

For certain copyleft licences, it can be difficult to achieve compatibility with copyleft licensing and receive the benefit of a patent licence, even a patent licence granted on RAND (reasonable and non-discriminatory terms),[67] although Red Hat has achieved it through its widely publicised agreement with Firestar. But Red Hat is in the unique position of having both the financial means and legal resources to negotiate such a licence.[68]

[65] See Section 7 of the GPLv2 available at <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html> accessed 19 March 2021. GPLv3, in contrast, has more limited restrictions upon further distribution in cases where the distributor has a patent licence allowing such distribution. See Section 11 of the GPLv3 available at <https://www.gnu.org/licenses/gpl-3.0.en.html> accessed 19 March 2021.

[66] Moglen, 'Free software matters: Patently controversial', see note 8.

[67] Discussed at length in Iain Mitchell and Stephen Mason, 'Compatibility of The Licensing of Embedded Patents with Open Source Licensing Terms' (2010) 3(1) *Journal of Open Law, Technology & Society (JOLTS)* 25–58 < https://jolts.world/index.php/jolts/article/view/57/100 > accessed 15 June 2022.

[68] See Red Hat press release, Red Hat Legal Team, 'Red Hat Puts Patent Issues to Rest' *Red Hat Blog* (11 June 2008) <http://www.redhat.com/about/news/archive/2008/6/red-hat-puts-patent-issue-to-rest> accessed 19 March 2021.

Often in cases of (corporate) patent litigation, the parties involved can and often do come to settlement through cross-licensing and patent peace agreements. These agreements are non-aggression agreements providing each party royalty-free access to a determined part of the other party's patent portfolio and often to specified products. This is prevalent in areas such as hardware manufacturing or biotech, and RF cross-licences are quite common in the computer hardware and software industry among proprietary companies. However, the nature of Open Source makes cross-licensing potentially non-viable; first, very few (if any) Open Source projects have any patents with which to 'trade' with a potential patent asserter. Second, there may not be a particular institution or entity with which to negotiate such an agreement—with the exception of corporate sponsored developments, such as Red Hat, which as we have mentioned, can and have negotiated patent licences; in addition, the GNOME Foundation recently negotiated a settlement of a patent assertion made against some of its Open Source.[69] Third, any potential legal entanglement due to software patents creates uncertainty and significant fear within the project community. Few Open Source projects are going to go near any patented technology or process—if they ever get to know about it—merely due to the risk of patent litigation and the transaction costs for dealing with the patent situation.

It has been argued, in the context of patents over standards, that from an economic perspective patent licences and royalties may be compatible with Open Source development models (this is fully discussed in Chapter 12): it is just a question of implementing an appropriate technological or business process for licensing and collecting the dues.[70] Indeed, there are Open Source projects such as Fluendo[71] whose very existence and business model lies in dealing with patents rights over audiovisual codecs, and interested third parties can purchase licences to these patent rights so as to implement and distribute proprietary patented codecs in Open Source multimedia environments. However, above and beyond the legal incompatibility when using copyleft licences, most non-commercial (and many commercial) Open Source projects are particularly incompatible with royalty-bearing technologies, since an essential characteristic of the project is to share the code easily among community participants (including users), and they have no visibility or control of downstream users. Requiring even minimal royalties would greatly hinder the freedom of developers to share and distribute the code they write.

This is reinforced by the sheer number of software-related patents that are applied for and issued annually (particularly in the United States), as well as the legal

[69] See section 10.6 later in this chapter.

[70] Jay Kesan, 'The Fallacy of OSS Discrimination by FRAND Licensing: An Empirical Analysis' (22 February 2011) *Illinois Public Law Research Paper* No 10–14.

[71] Available at Fluendo <http://www.fluendo.com> accessed 19 March 2021.

uncertainty about many of those that are issued (for lack of novelty, inventiveness, or patentable subject matter, as discussed earlier).[72] It would be impossible—if not counterproductive, as they could then be claimed to be knowingly infringing a patent, if subsequently litigated—for software developers to read through all the software patents relevant in their area of expertise (let alone 'all' software patents generally), and subsequently take an informed view on the validity, or not, of those patents.

Another significant area of concern for the Open Source community is the accumulation of patents in proprietary software companies. Usually, large companies like IBM use patents defensively. As they know that other companies in the industry will apply for patents, and then may sue for patent infringement in order to gain a competitive advantage, a company that wants to defend itself files for its own patents to use against its competitors. This either creates a massive patent war, such as that that has occurred in the mobile device industry,[73] or creates a *détente* or hold-off between the company and its competitors where each could sue the other in a similar way, so neither one does (and eventually they enter into cross-licensing agreements such as those mentioned earlier). However, members of the Open Source community have historically shown concern with large proprietary corporations asserting patent claims, directly or through associated patent assertion and licensing entities such as Intellectual Ventures,[74] to acquire a range of software patents that they can potentially use in the future to attack and try to restrict the development and distribution of Open Source software.

Finally, and this is linked to the previous point, we must mention NPEs.[75] These entities accumulate patents solely for the purpose of demanding patent royalties from third parties, and do not themselves 'practise' or implement their patents or for that matter conduct any business other than licensing and asserting their patents. They do not make, use, import, sell, or offer for sale anything that could be infringing, inoculating them against countersuits. There are a significant number of these entities, such as Acacia Research Group, or Intellectual Ventures, holding large portfolios of patents (Intellectual Ventures is alleged to hold over 30,000 existing patents).[76] While NPEs typically target their activities against the

[72] Ballardini, 'The Software Patent Thicket', 207, see note 32.

[73] Involving Samsung, HTC, Motorola, and Apple, among others. See Don Reisinger, 'A look back at the great Apple-Samsung patent war' *EWeek* (8 August 2014) <https://www.eweek.com/mobile/a-look-back-at-the-great-apple-samsung-patent-war/> accessed 19 March 2021.

[74] Dennis Crouch, 'Intellectual Ventures: Revealing Investors' *PatentlyO* (18 May 20122) <http://www.patentlyo.com/patent/2011/05/intellectual-ventures-revealing-investors.html> accessed 19 March 2021.

[75] See Brian Yeh, 'An Overview of the "Patent Trolls" Debate' (2012) Congressional Research Service, <https://sgp.fas.org/crs/misc/R42668.pdf> accessed 15 June 2022, for a good overview of this problem.

[76] Todd Bishop, 'Intellectual Ventures sues HP, Dell and others over patents' *Geekwire* (12 July 2011) <http://www.geekwire.com/2011/intellectual-ventures-sues-hp-dell-patents> accessed 19 March 2021.

products and services of commercial entities, in particular proprietary software companies with funds to pay for royalties, they have also targeted Open Source, both Open Source-based commercial entities such as Red Hat, who had to deal, for example, with Firestar,[77] and non-commercial Open Source foundations such as the GNOME Foundation, who had to deal with Rothschild Patent Imaging (see section 10.6 later in this chapter). As opposed to litigation against industrial entities, where (negatively) the threat of patent retaliation or (positively) the offer of a cross-licence may be made, it is nearly impossible to use such a strategy against NPEs, leaving only the expensive (prohibitively so, for Open Source communities) options of paying a royalty or challenging the validity or infringement of the alleged patents, or abandoning the allegedly infringing software altogether.

Thus, in the end the patenting regime for software serves to benefit nearly exclusively large (proprietary) software companies with economic resources to apply for, defend, and litigate software patents, potentially to the detriment of the Open Source communities who are behind many of the current innovations in information and communications technology (ICT), unless efforts are made to assist these communities with patent threats.

In summary, software patents are expensive to acquire and enforce, and outside most Open Source projects' economic capabilities. They are also considered philosophically, culturally, and politically anathema to many Open Source communities and their members, as a restriction on their innovation. In addition, there is a perception that many of the patents that represent a potential threat against Open Source may be of dubious validity, due to lack of novelty or inventiveness—particularly given the continued development of tests for abstractness in the US. Even when they appear to be acquired for 'defensive' or other altruistic purposes, there has been no guarantee against someone later 'weaponising' them for use in an offensive attack.[78]

This has led the Open Source community in many cases to reject the current legal regime whose uncertainty enables obtaining patent protection (in any form, even the allegedly 'highly filtered' protection granted by the EPO) for software, arguing on the one hand that the whole system is too expensive for Open Source projects and small software publishers to benefit from (if they wanted to) and, on basis of their own experience and that of the software industry as a whole, that copyright provides sufficiently strong protection for software and incentive to innovate and create more.

In a now often quoted memo, Bill Gates said in 1991: 'If people had understood how patents would be granted when most of today's ideas were invented, and had taken out patents, the industry would be at a complete standstill today.'[79] On this

[77] Floyd Marinescu, 'Red Hat Sued Over Hibernate 3 ORM Patent Infringement Claim' *Infoq* (30 June 2006) <http://www.infoq.com/news/RedHat-Sued-Due-to-Hibernate-3-O> accessed 19 March 2021, settled in 2008.

[78] Schultz and Urban, 'Protecting Open Innovation', see note 51.

[79] Bill Gates, 'Challenges and Strategy Memo' (16 May 1991) <http://en.swpat.org/wiki/Bill_Gates_on_software_patents> accessed 19 March 2021.

issue Richard Stallman stated in 2004: ‘Software patents are the software project equivalent of land mines: each design decision carries a risk of stepping on a patent, which can destroy your project. Because every such patent covers some idea and the use of that idea, which by giving monopoly on patents inhibits the development of software.’[80]

10.4 How Open Source Deals with Patents

We now turn to see how the community has reacted to and deals with the several interactions and friction areas between patents and Open Source, and the perceived patent threat.

The Open Source community’s actions in this respect can be divided into two types of action: preventive measures, to minimise the impact of software patents on software freedoms, and reactive measures, taking action to neutralise current patent threats to free software development.

10.4.1 Patent clauses in Open Source licences

The first and most ‘structural’ preventive measure to deal with software patents is the incorporation of patent-related terms in Open Source licences. As we noted in the introduction, an Open Source project’s community norms and guidelines are reflected in the chosen licence terms: they set out the rules for participation, in particular for contributing to and using the project software. The community has leveraged the licences to set out rules regarding patent grants and non-assertion among participants.

10.4.2 First-generation Open Source licences

The first generation of Open Source licences, particularly the permissive licences such as the BSD and X11/MIT licences, did not expressly mention patent rights, though based on the wording of the licences there are arguments that either an express or at least an implicit licence is granted.[81] Some legal writers believe that implicit patent licences are uncertain and not binding (in particular when there is

[80] Richard Stallman, ‘Fighting Software Patents—Singly and Together’ (2004) <http://www.gnu.org/philosophy/fighting-software-patents.html> accessed 19 March 2021.

[81] Van Lindberg, ‘OSS and FRAND’, see note 3; Peterson, ‘Why so little love for the patent grant in the MIT License?’, see note 5.

no consideration), giving rise to questions regarding their scope or duration, the impact of combing potentially patented software distributed under these licences with other programs or hardware, and the creation of derivative works, or that the licences licence copyright rights only and no patent rights are conveyed.[82] This is not a happy situation with regard to legal certainty for the Open Source community, and while these licences are still popular, contributors and users with concern about potential patent assertions, or who own significant patent portfolios and wish to have greater certainty about which part of their portfolio is being licensed, may eschew these licences in favour of more recent versions with explicit patent provisions.

Where a company did want to use one of these more permissive licences (Google Inc, in this instance, with regard to WebM VP8 video codec technologies), it added a patent licence grant and peace terms in an additional clause, tying the patent grant to its implementation of the patent claims.[83] The impact of this is twofold: the code that Google has distributed is effectively granted under the MIT licence, a recognised and standard Open Source licence permitting easy use and adoption, while users of Google's version of the code are given comfort and protection as regards claims with respect to patents that Google and other contributors may hold in the codec.[84] Enhanced versions of the MIT and BSD licences—the Universal Permissive Licence[85] and the 'BSD+Patent' licence,[86]were also created to take the basic framework of the MIT and BSD licences and add to it an explicit patent grant.

The GPLv2, first published in 1991, included wording directed to patents, with a stated aim of making GPL'd software redistribution incompatible with software patents rights assertion—either by contributors or licensees of contributors. GPLv2 does not have an express patent grant or non-assertion covenant. While a licence by the original creator cannot take away patent assertion rights of a third-party patent holder (rights to restrict distribution and use of a software that embodies the patent for example against payment of a royalty), what it can do is prevent the redistribution of the original software at all if such distribution under the terms of the GPL2 is prevented by patents encumbering the software; hence the name of Clause 7 of GPLv2, 'liberty or death':[87]

[82] Heather Meeker, *The Open Source Alternative* (Trenton, NJ: John Wiley and Sons, 2008); Kappos and Harrington, 'The Truth About OSS-FRAND', see note 3.

[83] Google's WebM, 'Additional IP Rights Grant (Patents)' <http://www.webmproject.org/license/additional> accessed 19 March 2019.

[84] This of course does not guarantee that 'all' potential patent rights in the codec are licensed, as Google may not hold all those rights.

[85] Open Source Initiative, 'Universal Permissive Licence' <https://opensource.org/licenses/UPL> accessed 30 August 2020.

[86] Open Source Initiative, 'BSD+Patent Licence' <https://opensource.org/licenses/BSDplusPatent> accessed 29 August 2020.

[87] This phraseology is based upon a famous speech in early US history. 'Give me liberty, or give me death!' Wikipedia <https://en.wikipedia.org/wiki/Give_me_liberty,_or_give_me_death!> accessed 29 August 2020.

> 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.[88]

The GPLv2 also forbids imposing any additional restrictions (such as patent encumbrances) on the rights granted by the licence to the recipients of the software. If a distributor does so, for example by asserting patent rights, their licence under the GPL is terminated. This effectively means that a patent holder who distributes a software program based on GPLv2 code, embodying one or more of its patents, may no longer assert those patent rights against downstream licensees who redistribute that program onwards,[89] or who incorporate the program in their own product. What is more, this has the effect that if a GPLv2 licensee does get a third-party patent licence to exploit the software, then to be able to redistribute it they must effectively ensure that all downstream licensees are covered. This was made explicit in GPLv3, published in 2007,[90] and Red Hat achieved this in its agreement with Firestar (with respect to one of its Open Source programs, called Hibernate).[91]

10.4.3 Second-generation Open Source licences

As Open Source software and the Open Source licensing model gained more popularity into the late 1990s, and as simultaneously it became clearer that software patents would be found in jurisdictions around the world to satisfy the requirements of national law, there developed a desire for Open Source licences with clear and

[88] GNU Operating System, 'GNU General Public License, version 2, (1991) <http://www.webmproject.org/license/additional> accessed 19 March 2021.

[89] The impediment on patent assertion is based upon the theory that the 'liberty or death' provision of GPLv2 includes an implied patent licence. Richard Stallman, 'Why Upgrade to GPL Version 3' Free Software Foundation (31 May 2007) <http://gplv3.fsf.org/rms-why.html> accessed 29 August 2020. The extent to which an implied patent licence would be found in GPLv2, and of what scope that licence would have, is an unresolved issue which led to a more detailed, express, patent licence being included in GPLv3.

[90] GNU, 'GNU General Public License' <https://www.gnu.org/licenses/gpl-3.0.html> accessed 19 March 2021.

[91] Maureen O'Gara, 'Red Hat Settles Patent Claims Against It' *DZone* (11 June 2008) <https://dzone.com/articles/red-hat-settles-patent-claims-> accessed 19 March 2021.

express terms around patent rights. One of the first organisations to take on this issue was Netscape, which was considering freeing its 'Navigator' web browser in 1998. That browser was released under the Netscape Public Licence[92] (later migrated into the Mozilla Public Licence 1.1), which included express patent provisions. Since that time, most newly created and OSI-approved Open Source licences also include an express patent licence grant of some scope.

The development of patent provisions in second-generation Open Source licences generally addresses two separate, but arguably related, issues. First, they grant an express patent licence to patent rights that the initial developer, or any contributor to the project, may have in their contribution. These express patent licences are in a variety of different forms, and each have differently expressed language, so determining exactly which patent rights are granted, and by whom, and for what, requires detailed analysis of the particular licence and the particular grant. Second, many—but not all—patent provisions in second-generation Open Source licences provide for defensive patent grant suspension (sometimes referred to as 'patent retaliation'), specifying conditions under which the express patent grant from authors or contributors is terminated or suspended in the event of a party that has received a licence initiating some form of patent litigation or other patent assertion with respect to the software.[93]

The ASF 2.0 License (2004)[94] provides a patent provision template that can serve as a model for an appropriate express patent licence grant, as well as a defensive patent grant suspension. The ASF 2.0 licence includes an express patent licence from each contributor to 'make, have made, use, offer to sell, sell, import, and otherwise transfer the Work'. This grant covers the contributor's contribution by itself, or when that contribution is combined with the software to which it is contributed. Similarly, the Mozilla Public License (MPL) 2.0 (2012)[95] contains an express patent grant covering the present and future patents rights of a contributor for the 'making, using, selling, offering for sale, having made, import, or transfer of either [the Contributor's] Contributions or its Contributor

[92] The Mozilla Foundation 'Netscape Public License 1.0' <https://website-archive.mozilla.org/www.mozilla.org/mpl/mpl/npl/1.0/> accessed 30 August 2020.

[93] The particular scope of the defensive patent grant suspension is important in evaluating whether a licence containing it may properly be considered an Open Source licence. Facebook, as one example, created a licence which included a defensive patent grant suspension provision that suspended the express patent grant in the event of any patent assertion against Facebook, whether or not that assertion related to the software licensed under that grant. This provision was roundly criticised as being non-reciprocal and was later withdrawn by Facebook. Sarah Gooding, 'Facebook to Re-license React after Backlash from Open Source Community' *WordPress Tavern* (25 September 2017) <https://wptavern.com/facebook-to-re-license-react-after-backlash-from-open-source-community> accessed 30 August 2020.

[94] The Apache Software Foundation, 'Apache License, Version 2.0' <http://www.apache.org/licenses/LICENSE-2.0.html> accessed 19 March 2021.

[95] Mozilla Foundation, 'Mozilla Public License Version 2.0' (MPLv2.0) <http://www.mozilla.org/MPL/2.0> accessed 19 March 2021.

Version',[96] and excludes deletions from, or modifications made to, the code, or combinations of the code with other software or devices, or the code in the absence of the contribution by that particular contributor.[97]

GPLv3 (2007) also has an express patent grant; Section 11 provides that '[e]ach contributor grants you [the user] a non-exclusive, worldwide, RF patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version'. The 'contributor version' is defined as '[any copyrightable work licensed] or a work on which [that copyrightable work] is based' which a copyright holder authorises use under the GPLv3 licence. 'Essential patent claims' in GPLv3 are defined as:

> all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, 'control' includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

The last point regarding 'control' is interesting, as in practice it permitted the flexibility for Red Hat to acquire downstream patent sublicensing rights from Firestar, so as to ensure valid onward GPL-based licensing of the Firestar patents to which Red Hat received a license. GPLv3 also allows alternative mechanism to allow a distributor of GPLv3 code to receive the benefit of a patent licence yet ensure that the source code remains available to the public.[98] GPLv3 has another patent-related requirement, drafted in response to a transaction between Microsoft and Novell,[99] which was designed to prevent unusually structured business deals believed to be a 'work around' to the concept of 'liberty or death'.[100]

[96] MPLv2.0, see note 95, Section 2.1(b). The 'Contributor Version' in this section is defined as 'the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution', similar to the way the Apache 2.0 licence covers combinations.

[97] MPLv2.0, see note 95, Section 2.3.

[98] 'If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients.' GPLv3, Section 11.

[99] Cath Everett, 'Inside the Microsoft-Novell deal' *ZDNet* (30 April 2007) <https://www.zdnet.com/article/inside-the-microsoft-novell-deal/> accessed 30 August 2020.

[100] This is reinforced by a paragraph in cl. 11 of the GPLv3 that provide for this very situation: If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorising them to use, propagate, modify or convey a specific copy of the covered

These clauses in each of the above-discussed licences, as well as many other licences approved by the OSI since 1998, ensure that users of software under these licences get the specified patent rights from the upstream contributors to the work. This does not mean that use of the software is free of patent risks, as third parties may have patent rights over the work and may not have granted the user any licence, and in many cases, subsequent changes made to the program after distribution by a patent holder may be unlicensed. Nevertheless, the user is protected from patent claims by the contributors, who—if the contribution is of original code—are usually the persons most likely to have any patent rights in that contribution.

10.4.4 'Patent defensive suspension' clauses

Patent defensive suspension clauses come in several different 'flavours', depending on the scope and conditions for triggering the clause. These provisions are often structured as a condition of the original licence grant—either just the patent grant, or all grants, including copyrights. Most are structured to protect the specific software to which a patent holder is licensed; a few against any suits based on patent rights over any software, not just the licensed software, although these broader provisions are now looked upon as non-reciprocally discriminatory and violative of Open Source Definition 5.[101] The provision may also revoke patent rights, or all rights granted under the Open Source licence. In Table 10.1 we will look at four licences, chronologically the MPLv2.0, Apache v2, EPLv2, and GPLv3.

What do these provisions achieve? On the one hand, as noted, Open Source participants using software under these licences have a certain degree of safety from

work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it'. The overall aim of this is to ensure a level playing field, and guarantee freedoms for the whole chain of licensees taking a copy of the code under the GPL. 'You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.' This provision was specifically aimed at the Microsoft/Novell transaction.

[101] OSD 5 states that an open source licence must have 'No Discrimination Against Persons or Groups'. Open Source Initiative, 'The Open Source Definition' <https://opensource.org/osd> accessed 30 August 2020. Any provision that takes licences away from entities asserting patents outside of the particular project to which a licence is granted is believed, by many, to violate this non-discrimination provision.

Table 10.1 Comparison of Defensive Suspension Clauses

Mozilla 2.0: Section 5.2	
If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under [the copyright and patent licence] Section . . . of this License shall terminate.	This clause is of a scope relatively common for defensive suspension clauses, although it does include a suspension not only of patent licences, but other licences as well in the event of a patent assertion against the software. Note that it does allow such an assertion in the form of a counter claim or cross-claim (a claim that is filed in response to an initial claim against the patent asserter), which may provide for some litigation strategy gaming tactics to retain the benefit of the licences but still assert patents against the software.
Apache 2.0: Section 3	
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.	The defensive suspension clause in Apache 2.0 licence is similar to Mozilla 2.0, although it only revokes potential patent right grants; it does not purport to terminate any copyright licence. It also does not exclude cross-claims or counterclaims, like Mozilla 2.0.
Eclipse Public License 2.0: Section 7	
If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) [the patent licence Section] shall terminate as of the date such litigation is filed.	This section is similar to Apache 2.0, in that it only suspends patent licences, and, like Apache 2.0, does not exclude cross-claims or counterclaims, thus preventing potential litigation strategies to preserve the licence grant while still asserting patents. Unlike the Common Public Licence, a predecessor of Eclipse which has been deprecated, it does not attempt to suspend patent licences for assertions against other software.
GPLv3: Section 10	
You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.	GPLv3 maintains similar 'liberty or death' provisions as its version 2, commented on in the introduction (now called 'No Surrender of Others' Freedom' clause), and includes this patent peace clause. Breach of this undertaking (not to initiate patent-based litigation with respect to the software in question) would mean breach of the licence, and revocation of all licence rights (both copyright, and patent) subject to the reinstatement provisions (e.g. if the litigation is withdrawn). GPLv3 does not limit this litigation against 'developers' but would cover litigation against 'any entity', similar to the Apache and Eclipse licences commented on earlier.

patent-related threats from upstream contributors as a result of the express patent licence grants, and downstream licensees as a result of the patent defensive suspension provisions; this provides a degree of 'patent peace' among community participants. The more participants involved in the community, particularly large patent-holding entities, the greater the peace, and all the more so if the licence is copyleft, and thus maintains the same licensing terms downstream and throughout the community of users. This contributes to the ideals of the Open Source community, of providing safe access to knowledge of Open Source technologies, and derivative works thereof, and freedom to innovate. 'Licensees and their sublicensees should not be able to benefit from Open Source while at the same time forcing the licensor to pay royalties for patents embodied in that very software.'[102] However, the scope of this protection does vary, and it is important (especially for patent-holding users or contributors to Open Source projects) to understand the scope of the express or implied patent licence clauses as well as defensive suspension provisions, and how they interact with their patent portfolios. These clauses may discourage patent holders from participating in communities, either because the patent licence grants are too broad—or too indeterminate—or because the defensive suspension provision curtails their ability to assert their own patent portfolio against entities towards whom they wish to maintain a strong patent position. An example of this is where a company drafted a modification to the MPL1.1 in order to protect their portfolio, rather than use the standard version of the licence,[103] or in the case where additional patent provisions have been appended to existing Open Source licences.[104]

In addition, there may be questions of validity of parts of these clauses. First, with regard to 'future' acquired patents and patent rights, and second, with regard to extending the benefits of the clauses to non-licensees, or extending the obligations either to future users (holding other patents) of the Open Source technologies, or future holders of relevant patents (for example, through acquisition), who may be able to argue they are not party to the original bargain. This may be a question of privity of contract, if licences are deemed to be contracts in this respect; though for licences that are considered to be unilateral authorisations (and not contracts), the provisions would only be effective against licensees (i.e. users) of the code, as a condition of the licence grant.

[102] Lawrence Rosen, 'Dealing with Patents in Software Licences' *Linux Journal* (1 January 2002) <http://www.linuxjournal.com/article/5575> accessed 19 March 2021.

[103] MXM Public license submission, OSI Review, 'For approval: MXM Public license' (8 April 2009) <https://lists.opensource.org/pipermail/license-review_lists.opensource.org/2009-April/000722.html>, commented on by Glyn Moody 'Should an Open Source Licence Ever Be Patent-Agnostic?' *Linux Journal* (9 April 2009) <https://www.linuxjournal.com/content/should-open-source-licence-ever-be-patent-agnostic> both accessed 19 March 2021.

[104] David Thompson, 'Reading the Fine Print in Facebook React's Open Source License' *White Source* (17 May 2017) <https://resources.whitesourcesoftware.com/blog-whitesource/reading-the-fine-print-in-facebook-react-s-open-source-license> accessed 30 August 2020.

10.4.5 Open Source as prior art, peer to patent, and defensive publication

Another way of the dealing with the negative impacts of software patents in a preventive manner is to help avoid those patents being granted *ab initio*. Although efforts to get software processes—CIIs—totally excluded (in theory and in practice) from patentability have largely been unsuccessful in most of the jurisdictions of the world, there have been other projects that have claimed to try to reduce, during the patent examination process, poor-quality applications on the basis of lack of conformity with the main requirements for patentability: novelty and inventive step.

One criticism of the patent application examination process, in terms of quality, is that patent examiners rely heavily on databases of issued patents and published patent application, and occasionally scientific publications, to discover prior art. This means that a significant amount, if not all, of previously published software and software-related documentation—both proprietary and Open Source—may not be taken into account during the prior art search stage of the examination process.[105]

Open Source as Prior Art was a project launched in 2005 as an initiative to enable Open Source software repositories to be considered during this prior art search stage, 'improving accessibility by patent examiners and others to electronically published source code and its related documentation as a source of prior art'.[106] Unfortunately, software in online repositories is not published in a manner that can easily be mapped against the way patent applications describe the claimed methods or processes. To ensure such software is taken into consideration, it needs to be time stamped, documented, and ideally categorised or described in a manner that can be searched. While this aim was laudable, in practice it has been found to be particularly difficult and time-consuming, so it seems the project is currently inactive.

In another attempt at improving patent quality, 'Peer to Patent' was a project launched by the US Patent Office (USPTO) together with New York University Law School, aimed at taking advantage of the software community to supply the USPTO with information and discussion relevant to assessing the claims of patent applications during the examination process, opening this process to public participation and 'community reviewing'. The goal of this project was to help third parties identify, submit, and rank prior art that is relevant to a patent application. The results of the initial phases of this project resulted in several patent applications being rejected or narrowed as a consequence of peer reviewing.[107]

[105] 'Do USPTO examiners search open-source codebases?' *StackExchange* (22 September 2012) <https://patents.stackexchange.com/questions/401/do-uspto-examiners-search-open-source-codebases> accessed 31 August 2020.

[106] The Linux Foundation, 'Open Source as Prior Art (OSAPA)' <https://wiki.linuxfoundation.org/osapa/start> accessed 19 March 2021.

[107] See results commented on by Andrea Casillas, 'Peer to Patent Pilot 2 Results' <http://www.slideshare.net/acasillas11/peer-to-patent-pilot-2> accessed 30 August 2020.

While there have been several criticisms,[108] and although the project was eventually discontinued, Peer to Patent has been seen as one of the factors leading to the creation of certain new processes for improving the quality of patents under the US America Invents Act of September 2011,[109] notably the possibility for third parties to file pre-issuance submissions,[110] prior art something similar to the observations phase of European Patent applications.[111] In addition, the Peer to Patent project has highlighted the need to take into account all prior art, not just in theory but also in practice, that is relevant to the patent applications that a patent office is reviewing: websites, journals, textbooks, software development, user manuals, and other non-patent databases. Community involvement and online discussion also helps find this information. This has a positive economic effect, as avoiding *ab initio* the granting of poor-quality and/or invalid patents is significantly cheaper than a re-examination or post-grant review processes, or invalidity procedures before the courts. Ideally, prior art submitted in this way would gradually reduce the ability of non-practising entities holding poor-quality patents to threaten Open Source projects.

As a third leg in the strategy for avoiding 'bad patents', *defensive publication* is coming to be seen as one of the most efficient and effective measures. IBM, for forty years, produced a publication of inventions, which it developed but did not seek to patent, as a mechanism for establishing prior art that might prevent others from later attempting to patent the same or similar technology.[112] Linux Defenders,[113] a program for defending the Linux operating system and the Open Source community as a whole against patent concerns and threats, and which also supported the Peer to Patent project, was an initiative to support defensive publication, by directing to a website 'Technical Disclosure Commons',[114] designed as a repository for individuals to make dated publications of technology disclosures for prior art purposes.

The Technical Disclosure Commons site provides a mechanism for developers and creators to submit a publication that is date-stamped so as to establish its public disclosure date for prior art purposes, with the goal that patent examiners and patent challengers may review and use these disclosures as prior art. The publications are posted to the IP.com prior art database, which allows patent offices worldwide to include these publications in their patent searches.

[108] Summarised at Wikipedia, 'Peer-to-Patent Criticisms' <http://en.wikipedia.org/wiki/Peer-to-Patent#Criticisms> accessed 30 August 2020.

[109] HR 1246 (112th), now Public Law 112–29, *Statutes at Large*, 125 Stat. 284 through 125 Stat. 341 (2012).

[110] 35 USC 122(e).

[111] WIPO has also taken up this idea for PCT applications, WIPO, 'Patent Cooperation Treaty (PCT) Working Group' (14–18 June 2010) <http://193.5.93.80/edocs/mdocs/pct/en/pct_wg_3/pct_wg_3_6.pdf> accessed 30 August 2020.

[112] 'IBM Technical Disclosure Bulletin' Wikipedia <https://en.wikipedia.org/wiki/IBM_Technical_Disclosure_Bulletin> accessed 30 August 2020.

[113] Linux Defenders: <http://linuxdefenders.org> accessed 30 August 2020.

[114] Technical Disclosure Commons <https://www.tdcommons.org/> accessed 30 August 2020.

10.5 Patent Busting and Patent Pools

The measures described earlier to make available and accessible more prior art so that those patents that are granted truly meet the tests of novelty and non-obviousness/inventive step are aimed at preventing the granting of poor-quality patents, not only in the software sector but in all technology fields. Another question for the Open Source community has been: what can be done about existing poor-quality patents that can be used to threaten the Open Source—and indeed proprietary software—communities and result in claims for unreasonable patent royalties or potentially injunctive remedies to stop distribution? This is a question of 'problem containment' and the strongest proposals so far focus on post-grant patent review, and creating defensive patent pools to protect specific areas of technology. Notably, these proposals are centred in the US, where the software and business method patent problem is believed to be most acute.

As regards patent review, there have been several community initiatives: one was the Linux Defenders project called 'Post-Issue Peer to Patent' which was designed to solicit prior art contributions from Linux and the broader Open Source community to permit the invalidation of previously issued patents that were issued in error because of the patent office's lack of awareness of relevant prior art. Like many of the initiatives discussed earlier, this project is no longer operational. Another is the 'Patent Busters' project, launched in 2004 by Electronic Freedom Foundation (EFF),[115] which organised collaborative community efforts to challenge existing patents that it had pinpointed as being particularly harmful to innovation. It then filed challenges to those patents it determined were not properly granted, which it had done with a certain degree of success.[116] The Patent Busters project does not appear to have taken any steps to 'bust' a patent since approximately 2016. The Public Patent Foundation (PUBPAT)[117] ran a similar project, with the aim of challenging through post-grant challenges US patents believed to be invalid. This project worked in all areas of technology, not just software,[118] although its activities have not been updated since 2015.

These actions have been supported in the US by the introduction in 2011 of post-grant review processes under the America Invents Act (AIA). One part of this legislation allows third parties to submit 'post-grant review' invalidity challenges

[115] EFF, 'Patent Busting Project' <https://www.eff.org/issues/patent-busting-project> accessed 16 June 2022.

[116] EFF at one time listed ten patents that challenged or wished to challenge under this project, with various degrees of success including complete invalidation ('busted'), narrowing, or some form of post-issuance reevaluation being initiated by the USPTO. Wikipedia, 'Patent Busting Project', <https://en.wikipedia.org/wiki/Patent_Busting_Project> accessed 16 June 2022.

[117] Public Patent Foundation, 'Undeserved Patents and Unsound Patent Policy Harm the Public' <http://www.pubpat.org> accessed 31 August 2020.

[118] Successes are listed at Public Patent Foundation, 'Protecting the public domain' <http://www.pubpat.org/Protecting.htm> accessed 31 August 2020.

of a recently granted patent, within nine months of issuance.[119] Grounds for invalidity include lack of novelty, obviousness, as well as non-compliance with description, enablement, or patent eligibility rules. Another part of this legislation allows third parties to submit '*inter partes* review' invalidity challenges at any time after 9 months from issuance, but only challenges for lack of novelty or obviousness based on patents or printed publications.[120] In the past, prior to the AIA, the only mechanism for patent challenges—the filing of an *ex parte* or *inter partes* re-examination—had to be based upon prior art patents or printed publications. The challenges represent a potentially cheaper mechanism for contesting the validity of an issued US patent than litigation, although the filing fees alone—over US $40,000.00—and the length and complexity of the procedures mean that such challenges can often mean costs in the hundreds of thousands of US dollars.[121]

One other mechanism to address patent threats is the creation of a defensive patent pool. The Open Invention Network (OIN),[122] controls a patent pool and has the mandate to defend Open Source—as defined in a Linux System Definition which began with core Linux but which today includes over 2,000 other Open Source packages—from patent attacks. It was launched in 2005, by six founding companies[123] and has received investment from four additional large technology industry participants[124] as well as its founders. OIN is free to join and works, at its simplest, on the basis of a mutual hold harmless, or commitment not to sue, amongst its 3,500 licensees, each of whom, like its founders, sign up to the same non-negotiable licence terms.

OIN has so far acquired a large (1,500+) portfolio of patents purchased at a cost in excess of US $100 million, 'all available royalty-free to any company, institution or individual that agrees not to assert its patents against the Linux System'. OIN will therefore buy patents (i) to stop them falling into the hands of non-practising entities, who might otherwise assert them against Linux-based companies;[125] and (ii) to provide a portfolio of patents that can be asserted against companies that attack Linux.[126] In fact, OIN partnered with Allied Security Trust to intercept

[119] 35 USC 321.

[120] 35 USC 311.

[121] Challenges to patents in the US based on prior art patents or printed publications using *ex parte* re-examination continue to be available, 35 USC § 302, and are likely much cheaper, but these proceedings can often be one-sided in favour of the patent holder and therefore generally are only used when the prior art is particularly strong.

[122] OIN <http://www.openinventionnetwork.com> accessed 31 August 2020.

[123] IBM, Phillips, NEC, Sony, Novell, and Red Hat.

[124] Canonical, TomTom, Google, and Nissan.

[125] See, e.g., OIN's purchase of twenty-two Silicon Graphics patents that Microsoft placed with Allied Security Trust to sell: Paula Rooney, 'OIN Outmanuevers Microsoft, Buys Linux Patents' *ZDnet* (9 September 2009) <http://www.zdnet.com/blog/Open Source/oin-outmanuevers-microsoft-buys-linux-patents/4800> accessed 31 August 2020.

[126] See, e.g., OIN's transfer of four patents to Salesforce.com after Salesforce.com was sued for patent infringement by Microsoft: Florian Mueller, 'The OIN gave Salesforce.com four patents to assert against Microsoft' *Fosspatents* (31 May 2011) <http://www.fosspatents.com/2011/05/oin-gave-salesforcecom-four-patents-to.html> accessed 31 August 2020.

Microsoft patents that were alleged to read on Open Source functionality and avoid those patents and associated claim charts from being 'washed' through Allied Security Trust (AST)—where they could have been licensed to AST's members before being passed to an NPE to have the claim charts enforced through litigation.[127]

Because all of the patents of all of the members of OIN are in effect licensed RF to all the other members in relation to the Linux System, that equates to a collective patent portfolio of over an estimated 350,000 patents and applications pledged not to be asserted against the Linux System software.

OIN acted successfully to convey patents from its extensive portfolio to Salesforce.com when it was sued for patent infringement of FAT filesystem patents by Microsoft. Rather than expose itself to a potential injunction, the counterclaim by Salesforce of the patents received from OIN precipitated a rapid settlement by Microsoft.[128] In addition, in at least one other action that has been made public, when TomTom was also sued by Microsoft over exFAT filesystem patents the spectre of OIN's conveyance of patents to TomTom coupled with TomTom's own patents that were used in the actual counterclaim was sufficient to trigger a settlement for a fraction of the original damage claim.[129] While there is little statistical data available regarding patent threats and assertions, the fact that OIN has routinely provided prior art to companies in the Open Source community at risk of, or actively in, litigation indicates that OIN's involvement may serve a useful vehicle to reduce patent threats in core Linux and the adjacent Open Source software space.

The foregoing NPE interventions notwithstanding, OIN historically was designed to primarily work to mitigate practising entity patent risk but since Microsoft became a member of the OIN Community in late 2018,[130] OIN has pivoted to put increasing focus on mitigating NPE risk. In addition to working with the Open Source technical community to identify prior art to be shared with Community members who are at risk or in litigation, OIN has also joined with the Linux Foundation, IBM, and Microsoft to found and fund the Unified Patents' Open Source Zone to enable the mitigation of risk from NPE-owned patents that read on Open Source functionality.

OIN is a defensive entity and not an assertion entity; that is, it has not itself commenced litigation against companies attacking Open Source using its existing patent portfolio, although OIN has sold hundreds of patents to companies in

[127] Nick Wingfield, 'Group of Microsoft Rivals Nears Patent Deal in Bid to Protect Linux', *Wall Street Journal* (8 September 2009) <https://www.wsj.com/articles/SB125236988735891147> accessed 19 August 2022.

[128] Florian Mueller, 'The OIN Gave Salesforce.com Four Patents to Assert against Microsoft', *FOSS Patents* (31 May 2011) <http://www.fosspatents.com/2011/05/oin-gave-salesforcecom-four-patents-to.html> accessed 19 August 2022.

[129] See comment by Software Freedom Law Center, 'Settled, But Not Over Yet' (30 March 2009) <http://www.softwarefreedom.org/news/2009/mar/30/settled-not-over-yet> accessed 31 August 2020.

[130] Navneet Akash, 'Microsoft Joins OIN, Makes 60,000 Patents Open-Source', *C#Corner* (12 October 2018) <https://www.c-sharpcorner.com/news/microsoft-joins-oin-makes-60000-patents-opensource> accessed 15 June 2022.

litigation or at risk from operating companies poised to assert patents containing Open Source-related claims.

In addition to OIN, it has been suggested that an assertion entity (Fair Troll) acting on behalf of the Open Source community to recoup sums paid in patent licensing might have attractions.[131] Given that the Open Source community in general has been vocally anti-software patent, creation of such an entity with community support seems unlikely.

In March 2013, Google published a proposal to establish and standardise defensive patent pools, with the objective of reducing patent litigation concerns, particularly by NPEs.[132] One particular part of this proposal that eventually came into fruition was the proposal of a Licence on Transfer (LOT) regime whereby companies would band together and commit that they would grant one another licences to their patents, even if those companies did not have in place any existing patent licence arrangements between them, in the event that one of their patents was sold or otherwise transferred to an entity that might be non-practising. This resulted in the formation of the LOT Network, in 2014, to achieve exactly this result.[133] The network has grown to over 1,000 participants in a relatively short period and, though the benefits of this network extend only to members and it is not an Open Source-specific solution in the manner of OIN, companies active in Open Source can gain protection from NPE risk by joining LOT; large companies pay a modest annual fee while companies below a certain size receive complimentary membership.

Finally, during the 2000s, various companies made patent pledges in favour of individuals and groups working on Open Source—unilateral promises not to assert patents against developers, provided that certain conditions are met. These pledges are intended to operate as an enforceable covenant not to sue, and equitable estoppel should preclude the patent holder from bringing suit against those within the safe harbour defined by the pledge.

Notable patent pledges include Red Hat,[134] Nokia,[135] and IBM.[136] Note also that many of these pledges may have been expanded by the joining of some of the

[131] Florian Mueller, 'The DPL and the "Fair Troll" business model: make money fighting patents with patents' *FOSS Patents* (18 May 2010) <http://www.fosspatents.com/2010/05/dpl-and-fair-troll-business-model-make.html> accessed 31 August 2020.

[132] Eric Schulman, 'Working together to reduce patent litigation' *Google Public Policy Blog* (12 March 2013) <http://googlepublicpolicy.blogspot.co.uk/2013/03/working-together-to-reduce-patent.html> accessed 31 August 2020.

[133] 'How We Protect Members' LOT Network <https://lotnet.com/how-we-protect-members/> accessed 31 August 2020.

[134] Promise at Red Hat, 'Statement of Position and Our Promise on Software Patents' <http://www.redhat.com/legal/patent_policy.html> accessed 31 August 2020.

[135] 'Nokia announces patent support to the Linux Kernel' Phys.org (26 May 2005) <https://phys.org/news/2005-05-nokia-patent-linux-kernel.html> accessed 31 August 2020. This pledge has a number of different qualifications, including '[w]ith respect to new functionality introduced into future Linux Kernel releases, Nokia reserves the right to declare that the Patent Statement shall not apply'.

[136] IBM, 'IBM Pledges 500 U.S. Patents to Open Source in Support of Innovation and Open Standards' (11 January 2005) <http://www-03.ibm.com/press/us/en/pressrelease/7473.wss> accessed 31 August 2020.

pledging entities to OIN—including Microsoft in October 2018, which has been followed by an extension of the Linux Definition to add the Microsoft exFAT patents into the Linux System Definition and OIN's pool.

One major unresolved issue is whether a pledge binds a new owner of a patent, an issue of great practical significance given the powerful and accelerating trend for major patent holders to divest some parts of their patent portfolio to patent assertion entities. This issue is also being considered in the context of whether FRAND obligations bind successors in title, as discussed in Chapter 11 .

10.6 Patent Litigations Initiated Against Open Source

Although concerns about the impact of patents against Open Source have been raised for at least thirty years (since GPLv2 identified patents as a concern in its preamble and the 'liberty or death' clause) and although numerous measures, as discussed earlier, have been implemented to address those concerns, actual threats (at least in the form of patent infringement suits filed against Open Source) have been surprisingly rare and generally resolved in a way favourable to the Open Source model. This data is somewhat contrary to the general trend of patent infringement litigation filings, which have shown, in the US, a steady-state of such filings by practicing entities, and a variable—but gradually increasing—trend of filings by NPEs.[137] Global trends also seem to indicate an increasing rate of patent infringement suit filings by NPEs.[138]

Although rare, there have been a few instances of patent infringement litigation filed against software licensed under an Open Source licence. In almost every instance, these litigations have been filed either ancillary to a separate, non-patent, dispute,[139] and in almost all cases, the patent is asserted against a for-profit entity that makes Open Source software part of its overall revenue-producing product profile.[140] In the one case where an actual verdict of patent

[137] See RPX Corporation, 'What 15 Years of U.S. Patent Litigation Data Reveal About the IP Market' *RPX Insights* (25 January 2021) <https://insight.rpxcorp.com/news/65081-what-15-years-of-us-patent-litigation-data-reveal-about-the-ip-market> accessed 20 March 2021.

[138] See Michael Crichton, Gregory Gramenopoulos, Vincenzo Jandoli, et al., 'Global Patent Litigation: Trends, Tools, and Strategies to Enforce Your Patent Rights Globally' *Strafford* (2 June 2020) <http://media.straffordpub.com/products/global-patent-litigation-trends-tools-and-strategies-to-enforce-your-patent-rights-globally-2020-06-02/presentation.pdf> accessed 20 March 2021.

[139] See, e.g., *XimpleWare v Versata,* Case No. 3:13cv5160 (N.D. Cal. 2013) (a copyright infringement action for failure to abide by GPLv2) and *XimpleWare v Versata,* Case No. 5:13cv5161 (N.D. Cal. 2013) (a corresponding patent complaint for patent infringement resulting from the failure to abide by GPLv2). These two suits, as well as other associated suits, were eventually settled. Sylvia Jakob, 'Versata saga settled with prejudice' *ifrOSS News* (19 March 2015) <https://www.ifross.org/?q=en/artikel/versata-saga-settled-prejudice-1> accessed 20 March 2021. Details of the terms of settlement are not public.

[140] See, e.g., *Bedrock Computer Technologies LLC v Softlayers Technology Inc.*, Case No. 6:09-cv-269 (LED) (E.D. Tex. 2009), which involved a patent infringement claim against Google, and others, for features in the Linux kernel. Google initially lost the claim and was assessed damages of U S$5,000,000. Steven Vaughn-Nichols, 'Idiotic Anti-Linux & Google Patent Decision' *ZDNet* (21 April 2011) <https://

infringement was found, and damages were assessed for that infringement, the verdict was rendered in summary fashion via a jury (thus not providing a detailed explanation of how the patent was infringed by the accused Open Source)[141] and was shortly thereafter settled and dismissed without explanation as to the terms of the settlement.

One of the earlier patent litigation assertions against Open Source software was a claim made against Red Hat's distribution of JBoss's Hibernate object-relational mapping tool (licensed under GPLv2) by the patent holder FireStar.[142] No court decision was rendered in that litigation, but upon settlement, Red Hat did make a statement assuring the Open Source community that that settlement was fully conformant with Red Hat's patent obligations under the 'liberty or death' provisions of GPLv2:

> The covered products include all software distributed under Red Hat's brands, as well as upstream predecessor versions. The settlement also protects derivative works of, or combination products using, the covered products from any patent claim based in any respect on the covered products. Essentially, all that have innovated to create, or that will innovate with, software distributed under Red Hat brands are protected, as are Red Hat customers.
>
> 'Red Hat's settlement satisfies the most stringent patent provisions in FOSS licenses, is consistent with the letter and spirit of all versions of the GPL and provides patent safety for developers, distributors and users of FOSS software,' said Richard Fontana, FOSS Licensing and Patent Counsel at Red Hat.[143]

A more recent patent litigation involving Open Source, which demonstrates the complex interplay of patent infringement assertions, the various mechanisms for challenging patents (both administratively and in court, in the United States), and continued controversy about the legitimacy of the mechanism for administratively challenging patents using Inter Partes Review (IPR) can be found in the activities of the patent holder Sound View Innovations (Sound View). Beginning in 2016, Sound View filed a series of patent infringement lawsuits—in

www.zdnet.com/article/idiotic-anti-linux-google-patent-decision/> accessed 20 March 2021. The case was eventually settled, with regard to Google, and dismissed, see Order Vacating Verdict and Dismissing Claims and Counterclaims (18 May 2011) available at <https://docs.justia.com/cases/federal/district-courts/texas/txedce/6:2009cv00269/116887/830> accessed 20 March 2021, although details of that settlement are not public.

[141] Jury Verdict, *Bedrock Computer Technologies LLC v Softlayers Technology Inc.*, Case No. 6:09-cv-269 (LED) (E.D. Tex. 15 April 2011) available at <https://docs.justia.com/cases/federal/district-courts/texas/txedce/6:2009cv00269/116887/746> accessed 20 March 2021.

[142] Paula Rooney, 'FireStar Files Suit Against Red Hat' *CRN* (7 July 2006) <https://www.crn.com/news/applications-os/190300990/firestar-files-suit-against-red-hat.htm> accessed 20 March 2021.

[143] Red Hat, 'Red Hat Puts Patent Issue to Rest' Red Hat Press Release (11 June 2008) <https://www.redhat.com/en/about/press-releases/patent> accessed 20 March 2021.

Delaware,[144] California,[145] and Colorado[146] in the US—accusing a variety of different companies, and a number of different technologies used by those companies, of infringing a portfolio of as many as seven US patents. At least some of these patents were claimed to be infringed by Hadoop data processing software (licensed under the ASF 2.0 Licence) and the JQuery JavaScript library (licensed under the MIT licence).[147] The Sound View patent asserted against Hadoop, US Patent No. 6,125,371 was eventually ruled invalid as the result of an IP filed against it, and that ruling was upheld by the US Court of Appeals for the Federal Circuit.[148]

Although non-commercial Open Source projects have historically avoided patent infringement suits, there is at least one, recent, incident of a direct assertion of patent litigation claims against an Open Source project itself, rather than a commercial entity making a business of distributing Open Source.[149] Rothschild Patent Imaging (RPI), an NPE associated with an inventor with a large number of patents held by many different NPEs, sued the GNOME Foundation's 'Shotwell' feature (licensed under LGPLv2.1) for patent infringement.[150] The patent lawsuit against the GNOME Foundation was ultimately settled with RPI granting a licence to all software—not just Shotwell, or GNOME code—licensed under an OSI-approved licence, without payment of any royalty, fee, or settlement amount, to any patent originating from the same inventor.[151] The scope of that settlement,[152] like the settlement with FireStar by Red Hat, may also have been driven by the 'liberty or death' patent provisions that, like in GPLv2, exist in LGPLv2.1. The Executive Director of the GNOME Foundation, Neil McGovern, expressed complete satisfaction with the ultimate resolution of that patent dispute: 'McGovern said he was 'exceptionally pleased with the outcome … I felt it was incredibly important to send a message to the entire patent assertion industry that basically you don't go

[144] *Sound View Innovations, LLC v. Facebook, Inc.* Case No. 1:16-cv-00116-RGA (D. Del. 2019).

[145] *Sound View Innovations, LLC v Hulu, LLC*, Case No. 2:17-cv-04146-JAK-PLA (C. D. Cal. 2017).

[146] *Sound View Innovations, LLC v Sling TV LLC*, Case No. 1:19-cv-03709-CMA-SKC (D. Col. 2019).

[147] Adam Philipp, 'Sound View Claims Open Source Software Infringes Patents' *AeonLaw* (22 May 2019) <https://aeonlaw.com/blog/2019/05/22/sound-view-claims-open-source-software-infringes-patents/> accessed 21 March 2021.

[148] *Sound View Innovations, LLC v Hulu LLC*, Case: 19-1865 (Fed. Cir. 2 July 2020).

[149] Campbell Kwan, 'GNOME faces 'baseless' patent lawsuit for organising images' *ZDNet* (26 September 2019) <https://www.zdnet.com/article/gnome-faces-baseless-lawsuit-from-patent-troll/> accessed 29 August 2020.

[150] Richard Speed, 'Fairytale for 2019: GNOME to battle a patent troll in court' *The Register* (25 September 2019) <https://www.theregister.com/2019/09/25/gnome_sueball_shotwell/> accessed 20 March 2021. Amanda Brock and Matt Berkowitz, 'GNOME

[151] Amanda Brock and Matt Berkowitz, 'GNOME Settles Litigation, Extends Patent Coverage to all Open Source Initiative Licensing' *The New Stack* (30 July 2020) <https://thenewstack.io/gnome-settles-litigation-extends-patent-coverage-to-all-open-source-initiative-licensing/> accessed 29 August 2020.

[152] The GNOME Foundation did not made the settlement agreement and licence terms public, and all mention of the lawsuit and settlement have been scrubbed from the GNOME Foundation's website. The settlement agreement was nevertheless posted by others, and can be found at the following location: <https://blog.hansenpartnership.com/wp-uploads/2020/09/GNOME_final_agreement_5-20_with_schedules.pdf> accessed 20 June 2022.

after open source projects. It won't end well for you.'[153] Whether the patent litigation against the GNOME Foundation represents an anomaly, or the start of a trend of NPEs asserting patents directly against projects themselves, remains to be seen.

10.7 Conclusions

The Open Source community attitude to patents has gone from raising the issue—rejecting software patents on principle—to implementing sophisticated mechanisms for dealing with them, both on a structural basis (in Open Source licences) and in public and private initiatives. Looking back at the initial objective of exploring the relationship between patents and Open Source, it can be seen that there are several areas of friction, creating risk and uncertainty. However, the different mechanisms mentioned that aim to reduce these issues are far from completing the task. What more can be done?

10.7.1 If you can't beat them … should you join them?

One view to take is that as the software patent system seems to be here to stay (in one form or another), the Open Source community should become a participant in the system if it wishes to protect itself from the threats of patent thickets, patent lawsuits, International Trade Commission (ITC) proceedings, patent-encumbered standards, and high awards in the event of infringement findings.[154] This means not only applying for patents and using them as to support defensive countermeasures (something in which Open Invention Network is actively involved) or aggressive measures, potentially creating patent pools for Open Source environments, but also providing Open Source technologies and ideas as searchable prior art and eventually taking a patent licence over Open Source technologies in terms that benefit the whole community, and which comply with copyleft licensing terms.[155]

However, this comprehensive approach is difficult in economic terms, considering the modest financial status of the great majority of Open Source projects

[153] Tim Anderson, '"This was bigger than GNOME and bigger than just this case." GNOME Foundation exec director talks patent trolls and much, much more' *The Register* (23 October 2020) <https://www.theregister.com/2020/10/23/this_was_bigger_then_gnome/> accessed 20 March 2021. After the settlement was made, a separate challenge to the Rothschild patent involved in the GNOME Foundation patent lawsuit was made in the USPTO, resulting in every claim in that patent being cancelled, thus reverting the subject matter of the Rothschild patent asserted against GNOME to the public domain. See OSI Staff, 'GNOME patent troll stripped of patent rights', *Voices of Open Source* (28 April 2022) <https://blog.opensource.org/gnome-patent-troll-stripped-of-patent-rights/> accessed 20 June 2022.

[154] See, e.g., the arguments of Schultz and Urban, 'Protecting Open Innovation', note 51.

[155] Two representative examples would be the settlements negotiated to resolve the FireStar against Red Hat, and the Rothschild assertion against GNOME, both of which were reported to have been settled under community-beneficial licence terms (see section 10.6 earlier).

(commercial or not), because they require substantial industry backing, such as the way in which the OIN and its various initiatives have the financial backing of significant market players such as Philips, NEC, Sony, IBM, Red Hat, Google, Toyota, and SUSE.

10.7.2 Patent reform

More recently, there have been a number of proposals for patent reform, the idea being that in the context of these conflicts, rather than forcing Open Source development to change and adapt its ways and methods (which have been proven to provide significant innovation and contribution to the 'Progress of Science and useful Arts')[156] to a legal framework that is unaligned with the functioning of the Open Source model, that instead the legal system itself that should be improved. Indeed, there are those that argue that the patent system in general has not led to greater innovation, especially in the field of software, as much as constituting a block on innovation and progress.[157]

Some writers have suggested significantly modifying the patent system, reducing the strength of patent protection, if not getting rid of patents altogether (at least for software), a view taken not only by the FSF[158] and the Foundation for a Free Information Infrastructure[159] but also some leading academics in the field.[160] Proposals include expressly eliminating or limiting software as patentable subject matter, tailoring the length of patent protection to software (to a period of much less than the current twenty years from first filing), or awarding patents only when

[156] US Const, Art I, s 8, cl 8, known as the patent and copyright clause.

[157] James Bessen and Michael Meurer, *Patent Failure* (Princeton, NJ: Princeton University Press, 2008), have found evidence that patents can actually harm innovation. Eric von Hippel concluded that 'empirical data seem to suggest that the patent grant has little value to innovators in most fields' in Eric von Hippel, *Sources of Innovation* (Oxford: Oxford University Press, 1988) available online at <http://web.mit.edu/evhippel/www/sources.htm> accessed 19 March 2021. In *The Wealth of Networks* (New Haven, CT: Yale University Press, 2006), Yochai Benkler suggests that patents may result in a drop in productivity. In Josh Lerner, 'Patent Protection and Innovation over 150 Years' (Nat'l Bureau of Economic Research, Working Paper No 8977, 2002), the author noted that strengthening available patent protection tended to yield less patenting of new innovations by domestic inventors, which may correlate with reduced rates of technological innovation.

[158] Richard M Stallman, 'Software Patents—Obstacles to Software Development' in *Free Software, Free Society: The Selected Essays of Richard M. Stallman* , see note 10.

[159] FFII <http://www.ffii.org>; and Stop Software Patents, 'Petition to stop software patents in Europe' <https://www.devroom.io/2010/01/19/sign-the-petition-stop-eu-software-patents/> both accessed 19 March 2021.

[160] Boldrin and Levine, *Against Intellectual Monopoly,* see note 52, conclude that 'a system that at one time served to limit the power of royalty to reward favoured individuals with monopolies has become with the passage of time a system that serves primarily to encourage failing monopolists to inhibit competition by blocking innovation' (at 20). See also James Bessen and Michael Meurer, 'The private costs of patent litigation' (Boston University School of Law Working Paper Series, Law and Economics, Working Paper No 07–08, online at <http://dx.doi.org/10.2139/ssrn.983736> accessed 19 March 2021), the authors conclude: 'In the worst case, the net effect of patents today may be to reduce the profits of public firms and to possibly impose disincentives on innovative activity as well.'

strictly needed on economic grounds (although the latter would be difficult given that most patent offices are ill-equipped to evaluate economic data).

Along similar lines, other more moderate changes have been proposed, to limit the effect of patents in the context of software. At a conference on Patent Reform at Santa Clara Law School,[161] Professor Mark Lemley, one of the leading advocates of patent reform in the US, suggested that the interpretation of US patent law should be tightened up, to prevent software patents from being drafted in general functional terms (thus prohibiting any implementation of the functional idea, creating an overbroad patent), and limit enforceable claims to the actual algorithms disclosed by the patentees and their equivalents. This rule is something is argued that the courts in the US should be doing under the Patent Act of 1952,[162] increasing disclosure obligations for software related patents and details of computer implemented functional claims, obliging applicants, for example, to use diagrams, flowcharts, or pseudocodes along with a clear description of the invention in natural language, and reducing the abstract nature of claims. This idea is also of some interest to the European Patent Convention regime, which generally allows functional claims but only to the extent that any more precise definition would reduce the scope of the invention (which is in fact the very purpose of ruling out functional claims).[163] The EPO Guidelines develop this, prohibiting attempts to define an invention purely in terms of the result to be achieved (thus claiming the underlying technical problem), particularly if a claim is formulated in such a way as to embrace other means, or all means, of performing the function.[164]

Another suggested idea is not to attack the upstream source of the problem, the patentability of software, which is proving to be fairly immutable,[165] but to limit the effect or enforceability of software patents on the market, reducing the liability risk for Open Source projects and users. One proposal is to legislate a 'safe harbour' from patent claims for software that runs on 'general purpose machines' (PCs and servers, terminal and mobile devices such as smart phones, routers, and set-top boxes, and so forth).[166] This may seem rather conservative, for example it would not apply to specifically programed hardware devices, and doesn't really deal with existing patents (unless the effect would be retroactive with regard to

[161] Santa Clara Law, 'Solutions to the Software Patent Problem' (16 November 2012) <https://law.scu.edu/hightech/2012-solutions-to-the-software-patent-problem/> accessed 19 March 2021.

[162] US Patent Act—35 USC, Article 112. See Mark Lemley and Julie Cohen, 'Patent Scope and Innovation in the Software Industry' (2001) 89 *California Law Review* 1. See also Ballardini, 'The Software Patent Thicket', see note 32.

[163] Article 83 EPC. See *Synergestic herbicides/CIBA GEIGY* T68/85, and subsequent cases.

[164] EPO Guidelines, C-III, 4.10 and 6.5.

[165] See the summary of the debate around software patents that occurred in the EU in Free Software Foundation-Europe, 'Software Patents in Europe', *FSFE Activities* <https://fsfe.org/activities/swpat/swpat.en.html> accessed 20 June 2022.

[166] Richard Stallman, 'Let's Limit the Effect of Software Patents, Since We Can't Eliminate Them' *Wired* (1 November 2012) <http://www.wired.com/opinion/2012/11/richard-stallman-software-patents> accessed 19 March 2021.

issued patents). Another suggested approach is to focus on interoperability and standards and only allow software patents to be enforced against implementations of standards where the patents had been previously declared during the standard setting process. 'All other software contexts should become off-limits for patent enforcement.'[167]

In the absence of any reform—a prospect that the Open Source community has advocated for more than twenty years but has never come close to fruition—Open Source projects must resort to classic defence strategies to deal with patent risks: obtaining a licence, proving non-infringement, proving invalidity due to lack of novelty, obviousness/lack of inventive step, or inventiveness (or requesting review, on the same bases), getting legal opinion support for invalidity or non-infringement (to reduce claims of wilful infringement), looking for other grounds for non-enforceability such as expiry, and eventually, of course, the technical solution of designing around the patent.[168]

[167] Simon Phipps, 'Stop patent mischief by curbing patent enforcement' *Infoworld* (9 November 2012) <http://www.infoworld.com/d/open-source-software/stop-patent-mischief-curbing-patent-enforcement-206658> accessed 19 March 2021.

[168] See Richard Fontana et al, 'A Legal Issues Primer for Open Source and Free Software Projects' *Software Freedom Law Center* (2008) <http://www.softwarefreedom.org/resources/2008/foss-primer.html> accessed 19 March 2021.