GPL-3.0 in the Chinese Intellectual Property Court in Beijing

Lucien C.H. Lin,[a] *Navia Shen,*[b]

(a) Legal Adviser, Open Culture Foundation; (b) IP Counsel, Huawei Technologies Co., Ltd.

DOI: 10.5033/ifosslr.v10i1.126

Abstract
With the increasing use of Free and Open Source Software (FOSS) in the world, the licensing issues and disputes regarding such licenses have been litigated in various jurisdictions. In the past, these lawsuits were concentrated in Europe and the United States, but less so in the Asia Pacific region. However, in 2018, the specialized Intellectual Property Right Court in Beijing, China, acting as a court of first instance, issued a decision in a software copyright infringement lawsuit related to FOSS. The defendant chose to invoke the copyleft mechanism in the GNU General Public License 3.0 (GPL-3.0) license as a defense against claims of copyright infringement. Although the court did not directly interpret the GPL license at this stage, the decision strongly implies that the GPL and the other FOSS licenses can be treated as valid in China. Even so, quite a number of details regarding the use of the GPL in China still require clarification, included as to how the license can substantially be enforced and implemented.

Keywords
Copyleft, GPL, derivative work, copyright infringement

Although most of the academic opinions are positive,[1] many commentators and practitioners did have doubts about whether a Free and Open Source Software license written in English could be enforced legally in China. After all, in 2014 the China Open Source Software Promotion Union (COPU)[2] once published a draft of "COPU Open Source General License Agreement V.1.0".[3] The text of COPU 1.0 was written purely in Simplified Chinese language and was meant to be used as an alternative solution in China for Chinese Free and Open Source Software projects. The COPU 1.0 actually was not used in any released Free and Open Source Software project due to the resource limitation for project development, and this license lasted only at the stage of public comments. However, the draft and publication of the COPU 1.0 reflected concerns as to whether Free and Open Source Software licenses written in foreign languages could be enforced in full in China without

1 As discussed in YANG XIA, *Introduction to Software Protection under Chinese Law*, http://ifosslawbook.org/china/, Section "Analysis of FOSS Under China Law".
2 http://www.copu.org.cn/about [retrieved June 2018]
3 https://www.oschina.net/news/52060/coup-license-comment [retrieved June 2018]

obstacles. Back to 1991, it was stipulated in the "China Regulation on Computers Software Protection", article 18, that in the case of a license to exploit software copyright, the license shall be made in formality according to the related laws and regulations of the China government. This requirement for formality has been removed in the revised version of the regulation, however some people still have doubt that whether or not a software license or contract not written in Simplified Chinese language could be fully applied in disputes during trial proceedings. This doubt was one of the reasons that the COPU group, supported by the China government industrial administration departments, tried to to prepare a new FOSS license suite purely written in Simplified Chinese on their promotion activities. This doubt remained, but now it seems to have been answered in the recent case of DCloud vs APICloud.[4] The plaintiff in this lawsuit is Digital Paradise (Beijing) Network Technology Co., Ltd. (DCloud), and the defendants are Pomelo (Beijing) Technology CO., LTD. & Pomelo (Beijing) Mobile Technology CO., LTD. (APICloud). The case, involving civil software infringement litigation, was filed in 2015 and a decision was handed down in April 2018. In this lawsuit, the GNU General Public License version 3 (GPL-3.0), especially the copyleft mechanism in it, was reviewed by the trial judges of the trial bench. The decision of the court affirmed the enforceability of the license.

Plaintiff's Claim

The plaintiff DCloud asserted that in September 2014 the defendant APICloud copied and adapted three independent plug-ins of plaintiff's HBuilder software development kit into the defendant's released APICloud toolset. The registered names of the allegedly infringed plug-ins in order in National Copyright Administration of China were "CIM plug-in", "ACR plug-in", and "HTML code drawing in real time plug-in". The plaintiff alleged it was the copyright owner of the HBuilder software, and that HBuilder was developed and largely released as shareware for limited use at no charge. While some of the modules and plug-ins in the HBuilder project were provided under certain FOSS licenses, including the GPL, these three allegedly infringed plug-ins were independent software works not provided under FOSS license. As such, the allegedly unauthorized copying and distribution of these three plug-ins infringed the right of reproduction, the right of alteration, and the right of information network dissemination protected under Article 9[5] of the Copyright Law of the People's Republic of China (2010 Amendment). Based on that, the plaintiff sued for the judgment of the court, demanding that the defendants publish an apology statement on its website www.apicloud.com and also on the other appointed information platforms for one month. Other than that, plaintiff also demanded RMB 3.5 million as compensation for copyright infringement, economic losses and legal costs.

Defendants' Defense

The defense of Pomelo (Beijing) Technology CO., LTD. & Pomelo (Beijing) Mobile Technology CO., LTD. (APICloud), as the defendants, was that part of the modules and plug-ins in the HBuilder project released by the plaintiff were derived from previously existing GPL-3.0-licensed components, such as "Aptana"[6] originally developed by Appcelerator, INC. under GPL-3.0 as a module in the Eclipse framework. Therefore, HBuilder project should be considered open source software made available under the GPL-3.0 license, and anyone has the right to use the code and create derivative works based on it under the terms of the GPL 3.0 license. Under this understanding of GPL-3.0, defendants asserted that plaintiff's consent was not required to use parts of the source codes from the HBuilder project for the APICloud project, and this kind of usages of software licensed under GPL-

4 (2015) 京知民初字第 631 号 / (2015) Jingzhi MinchuZi No. 631 of 22/03/2010 http://www.bjcourt.gov.cn/cpws/paperView.htm?id=100734294859&n=1 [retrieved Jan. 2019]
5 http://www.lawinfochina.com/Display.aspx?lib=law&Cgid=127326#menu1
6 https://github.com/aptana/studio3

3.0 should not constitute infringement of copyright. In addition, even if the disputed activities constituted infringement, the compensation requests have no facts or legal basis: APICloud project and DCloud project are both provided for free, the three disputed plug-ins are not core software of plaintiff, only minor parts of DCloud project are used, and defendants exhibited no subjective malice. Moreover, defendants asserted that there was no legal basis to demand publication of an apology statement. On account of the reasons above, the defendants requested that the court dismiss the plaintiff's claim.

Court Forensics and Judgement

The facts and legal judgements of the court in this case focus on copyright substantial similarity and forensics determining the relationship between the software. The identification task was entrusted to the Judicial Authentication Institute for IP Rights of CSIP.[7] Based on its analysis, the Authentication Institute reported:

> *On the first phase of the identification work required by the claimant, between the source codes of HBuilder and APICloud on plug-ins with the same or similar functions, for the CIM plug-in, there are 29 of the 30 source code files in the APICloud project being identified as substantially similar to the HBuilder project. For the ACR plug-in, 18 of the 23, and for the HTML code drawing in real time plug-in, 44 of the 56.*

> *Then on the second phase of the identification required by the defendants, the source code files found similar between HBuilder and APICloud, once more were verified with the third party's and Free and Open Source Software components prior to the release date of HBuilder provided by the defendants, for the CIM plug-in, there is none of the 29 source code files being identified as substantially similar to the previous Free and Open Source Software components. For the ACR plug-in, 13 of the 18, and for the HTML code drawing in real time plug-in, 2 of the 44.*

In accordance with the reports of the forensics above, given that 13 of the 18 between the ACR plug-in and the Free and Open Source Software components are similar, one might argue the GPL derivative issue for the ACR plug-in can be studied further, however, the judges of the trial bench ruled in the written judgment that "Of the aforementioned source code of similarity, only a small part of the source code is the same as the third-party or Open Source Software provided by the defendants." Hence, the conclusion by the court (discussed further below) is that the three plug-ins in dispute are independent copyrighted works of plaintiff, not derivative works of GPL-licensed software, the court of trial held that defendant infringed plaintiff's right of reproduction, the right of alteration, and the right of information network dissemination protected by the Copyright Law of the People's Republic of China. Therefore, the court ruled that the copyright infringement shall be compensated in the amount of RMB 1.25 million in economic losses and RMB 39,480 in lawsuit costs.

The Crucial Point

The crucial point of this lawsuit is that the defendants have proposed the copyleft mechanism in the GPL-3.0 as their primary defense method by claiming that the HBuilder project as a whole should be made publicly available under the GPL-3.0 license, and also alleged that their modification from the HBuilder project to the APICloud project are lawful acts permitted by the GPL-3.0 license. As for the GPL-3.0, the court of trial did not, in principle, deny the validity of it as a license agreement

7 Judicial Authentication Institute for Intellectual Property Rights at China National Software and Integrated Circuit Promotion Center (CSIP) of Ministry of Industry and Information Technology, at: http://www.csipsfjd.org.cn/

during the whole trial process. The court even introduced many paragraphs of the GPL-3.0 license in the written judgment for the factual section, for example, these contents of the GPL-3.0 have been translated into Chinese and quoted in the legal reasoning:

> *0. Definitions.*
>
> *"The Program" refers to any copyrightable work licensed under this License.*
>
> *[...]*
>
> *To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.*
>
> *5. Conveying Modified Source Versions.*
>
> *You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:*
>
> *[...]*
>
> *c) […] This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged […]*
>
> *d) [...]*
>
> *A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.*

Overall, the court of trial supported the validity and enforceability of the terms of GPL-3.0 and seemed to be willing to issue a decision based on the relevant provisions of the GPL-3.0. The main reasons presented by the court of trial in the written judgement are:

1. Based on the two identification results, the three plug-ins in dispute among HBuilder project and APICloud project do have quite a number of similarity issues of source code citation and modification, and only small parts of those similar source code have similarity issues with previous third party and other Free and Open Source Software. And for that reason, the court held that APICloud has copied and modified those plug-ins of HBuilder project for defendant's APICloud project.

2. Based on the copyright registration certificates for those three plug-ins, and plaintiff's explanation, the court held that plaintiff is the copyright owner of those three plug-ins, and those three plug-ins are separate and independent works and can be executed independently. This finding was based on

the fact that there is no GPL license text in the subdirectories of the three plug-ins or in the root directory of the HBuilder project. Although one other subdirectory of HBuilder contains GPL license text, the court held that that license text does not apply to the three plug-ins in dispute. Furthermore, the court held that all the three plug-ins are not derivative works or modifications referred to in the GPL license, which would have required the source code of the plugins to be made available publicly under the GPL license.

3. Based on above 1 and 2, the court further held that defendants' defense that Claimant's software shall be Free and Open Source Software was not supported. As such, the court held that defendants infringed copyright owner's rights of copying, adaptation and information network dissemination.

Judging from the grounds of judgement above, this decision made in this first instance can still be reasonably appealed to a higher court. However, if the defendants can't substantiate that the three plug-ins in disputes are derivative works of GPL licensed software rather than independent works, such as by deeply analyzing the interaction relationship between the GPL licensed parts and the other parts, including the three plug-ins in dispute, as well to assert that license text is not attached doesn't avoid corresponding codes for the derivative works to be made available publicly under GPL license. Even if the appeal is allowed, the defendants still have much to do to turn the tide in the followed proceedings. Usually the rulings of the Beijing IPR court are based on the reliance and respect for the forensics made by the CSIP. That means if APICloud can't make a credible argument regarding the copyleft effect for the appeal, both in legal inference and technical analysis for explaining why the original judgment is in contravention of the laws and regulations, their appeal might be treated as meritless and not favored by the trial court on appeal. Still, if those evidences are successfully substantiated, it will make the appeal case to be very complicated, as the court would be required to determine what constitutes a derivative work under GPL license and, if software is considered a derivative work of GPL-licensed software, then whether or not the defendants can directly procure and use these source codes under GPL license without additional permission of the Claimant as they asserted, and whether the defendants can require the Claimants to provide the related source code under the GPL.

According to the online article[8] published by the plaintiff's attorney in this case, although the defendants proposed to invoke the copyleft mechanism of GPL-3.0 as its defense, the arguments of the APICloud group were weak and not persuasive. That is, the defendants neither can explain what is their interpretation for the copyleft mechanism of GPL-3.0 in detail, nor can respond properly to the distinction between covered work as a whole and aggregation as separate parts in a compilation solution proposed by the plaintiff. In brief, assuming that the Hbuilder software contained some GPL 3.0 software, the court could either have viewed the Hbuilder software as subject to the GPL 3.0 license as a whole or instead as an aggregate not subject to the GPL 3.0 license. In this lawsuit, since the involved plug-ins are treated as separate works not based on prior GPL 3.0 software according to the entrusted forensics, the burden of persuasion fell upon the defendants, and the defendants failed to persuade the judges in court their way is the right way to do the copyleft interpretation, the judges made the final decision on the side of the plaintiff.

In Conclusion

In comparison with other international Free and Open Source Software litigation, this verdict does not provide much further analyses and in-depth explanations of how the Free and Open Source Software licenses should be evaluated and enforced in judicial proceedings. However, from a symbolic point of view, this case does have the value of being recorded and tracked. The main

8 *Will your cheese be taken away on account of Open Source licenses? - The constitution of copyright infringement of computer software involving open source licenses*: http://www.unitalen.com.cn/html/report/18040838-1.htm [retrieved June 2018]

reason is that the Beijing Intellectual Property Right Court is a specialized court in the intellectual property right field, the presiding judge and the other two People's Assessors in this trial, comfortably showing their support for the validity of GPL-3.0 without raising any doubt or objection. The disputed plug-ins in this ruling such as CIM plug-in, ACR plug-in, and HTML code drawing plug-in alledged as copyright infringements by the plaintiff are deemed to have no copyleft issues based on the CSIP forensics in the conclusion. However, because the defendants claimed the copyleft mechanism as their defense in the early stage, for the first time, the differences between a “covered work” and an “aggregate” for the Modified Versions of the Programs licensed under GPL-3.0 have been introduced by the Beijing IPR court. This lawsuit can be regarded as the beginning of judicial interpretation of Free and Open Source Software licenses in China.

As a matter of fact, the APICloud group, as the defendants of this case, have already made a positive statement[9] that they are appealing to the higher court for the second instance. In this statement, the APICloud group did admit that due to the lack of due diligence, back to 2015, when part of the plug-in codes from the HBuilder project were imported into the APICloud project, they didn't do it very well on filtering out the third party modules with no Free and Open Source licensing notice. However, after the dispute occurred and was notified by the DCloud in the same year, they subsequently released a new version of the APICloud project, which all has been licensed under GPL-3.0, and provided publicly to anyone on the hosting page of APICloud project onto GitHub[10]. By now, the APICloud group still believe that on account of the application and interaction method to the original GPL-3.0 modules in the HBuilder, the HBuilder project as a whole should be made available under GPL-3.0 without a difference. Therefore, more distinction and clarification for the covered scope of GPL-3.0 in the scenario of derivative or adaptation will likely be further discussed in the legal proceedings to come, and the subsequent effects and impact are worthy of continuous observation.

About the authors

***Lucien Cheng-hsia Lin**, legal adviser both of Open Culture Foundation and Gemly Int'l Intellectual Property Right Office, has been participating in the Open Source, Open Data, and Creative Commons Licenses interpretation and clarification among the local communities, official agencies, and companies in Taiwan for more than 10 years. He is best known for being the main proposer and drafter of the "Open Government Data License Taiwan 1.0" (https://data.gov.tw/license), with an one-way CC BY 4.0 switching mechanism implemented, which can make most of the materials on Taiwan Open Data portal available under CC BY 4.0 license.*

***Navia Shen**, legal counsel of Huawei Technologies Co., Ltd, has been working in Huawei for copyright and open source related affairs for about ten years.*

9 https://community.apicloud.com/bbs/thread-86486-1-1.html [retrieved June 2018]
10 https://github.com/apicloudcom/APICloud-Studio [retrieved June 2018]

Opposing the Monetization of Linux: *McHardy v. Geniatech* & Addressing Copyright "Trolling" in Germany

Marcus v. Welser [a]

(a) Dr. Jur., LL.M.; Attorney-at-Law, Vossius & Partner, Munich, Germany;, Certified IP Attorney; Lecturer at Humboldt-University, Berlin, Germany.

DOI: 10.5033/ifosslr.v10i1.128

Abstract
Over the past several years, many companies have received warning letters in Germany for GPL non-compliance from a particular programmer – Patrick McHardy. In these letters, the programmer regularly claimed to own copyrights in parts of the Linux kernel and requested that the addressees sign cease-and-desist declarations – subject to the payment of contractual penalties to him personally, in the event of future instances of non-compliance. This article describes court proceedings in Germany opposing the programmer's efforts, why Germany has been the venue of choice for these sorts of non-compliance assertions, how one particular company successfully defended itself, and discusses how other entities accused of non-compliance using these arguments can also pursue a successful defense.

Keywords
Law; information technology; Free and Open Source Software; GPL; GNU General Public License; Copyright Enforcement; Linux; McHardy; Copyright Trolling; Litigation.

I. Introduction; Statutory & Factual Background

Over the past several years, many companies in the electronics industry have received warning letters from a programmer – Patrick McHardy – who has claimed that he owns copyrights in the Linux kernel, and which he claimed were being infringed by these companies as a result of their non-compliance with the terms of the GNU General Public License, version 2 (GPLv2). He has regularly requested that the companies to whom he has sent these letters sign a cease-and-desist declaration, which includes a contractual penalty – payable to him personally in his capacity as alleged copyright owner – in the event of future allegations of breach of the GPL.

The Linux kernel is licensed under the GPLv2, which grants a license to the respective company distributing the Linux kernel. Under German copyright law, the exploitation rights granted under

GPLv2 are subject to a resolutive condition pursuant to Section 158 (2) of the German Civil Code.[1] If GPLv2 is not complied with – a condition subsequent in that license under German law – the right to use is automatically terminated and the prior legal situation is restored, where the user has no license.[2] In the case of distribution of software without complying with the conditions set forth in GPLv2, under German law, the copyright owners are entitled to make a cease-and-desist claim.[3] The copyright owner can enforce the right to cease and desist in court by applying for a permanent injunction and/or a preliminary injunction. In the case of a copyright infringement assertion, an injunction prohibits continued infringement. A permanent injunction is issued by a court after the main court proceedings. A preliminary injunction, on the other hand, is an interim measure which can be obtained in urgent cases within a few days or even hours. If the accused infringer does not comply with an issued injunction, the copyright owner has the right to file with the court a request for a coercive fine. Such a coercive fine may amount to a maximum of € 250,000, for each single case of future violation, although the courts typically will start with an amount lower than € 250,000. If the violation of the injunction continues, the amount of the fines will be raised for every incident.

It is within this factual and legal framework that the programmer McHardy has engaged in a pattern of using his contributions to the Linux kernel[4] to assert cease-and-desist rights against GPLv2 license violations and to extract settlement agreements obligating the initial violator to contractual penalties for future violations. As long as these settlements were never opposed or challenged through the German court system, this strategy resulted in numerous companies signing such settlement agreements – only to be faced with contractual claims based on allegations of subsequent violations.

Mr. McHardy's attempt to create precedent favourable to this strategy recently failed before the Higher Regional Court of Cologne. The Respondent in the proceedings in Cologne – Geniatech Europe GmbH – received a warning letter from McHardy, of the type that many companies had received from him previously, claiming non-compliance with GPLv2 for the Linux kernel, and therefore a violation of McHardy's alleged copyright rights in that program. Subsequently, Geniatech put their products into complete GPL compliance – without delay and with the help of technical experts knowledgeable about ensuring compliance. However, Geniatech neither signed the cease-and-desist declaration requested by McHardy nor paid any monetary sum to him. Therefore, McHardy applied for a preliminary injunction at the Regional Court of Cologne, i.e. the Court of First Instance. The Regional Court of Cologne issued a preliminary ex parte injunction in McHardy's favour.

Due to the fact that the court granted the decision without hearing Geniatech in advance, Geniatech filed an opposition and an oral hearing was scheduled. Geniatech, inter alia, argued that McHardy was not entitled to enforce the claims raised since he was not a joint author of the Linux kernel, and that his business model constitutes an abuse of law. However, the Court of First Instance did not reconsider the case, but instead confirmed its preliminary injunction after the oral hearing. Geniatech appealed the decision.

During the oral hearing, the Appeal Court – i.e., the Higher Regional Court of Cologne – cast doubt that McHardy could be considered a joint author under German copyright law of the Linux kernel, or even of the Linux kernel Netfilter component. In response to these questions about authorship raised by the Court, McHardy withdrew his request for the issue of a preliminary injunction against Geniatech.

1 German Civil Code (Bürgerliches Gesetzbuch, BGB), § 158(2) (stating that if a legal transaction is subject to a condition subsequent, the effect of the legal transaction ends when the condition is satisfied and the previous legal situation is restored).
2 *Id., see also* GPLv2, § 4 ("You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.") https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html
3 Higher Regional Court of Hamm, Judgment of June 13, 2017, File No.: 4 U 72/16.
4 In the case *McHardy v. Geniatech,* McHardy claimed to have contributed to the Linux kernel Network Stack and to Netfilter.

II. Germany as an Attractive Forum for Enforcing Rights in GPL Licensed Software

The fact that McHardy focused his enforcement activities in Germany is no accident. Many of the companies which had received his warning letters are multinational companies with global product distribution – such as manufacturers of smartphones – including jurisdictions such as China or the United States, where the distribution volumes would be significantly higher than in Germany. The reason why it was Germany which was chosen as the place to raise proceedings relates to particular procedural aspects of German law which are favourable to those enforcing intellectual property rights, but also can be seen as encouraging abuse of enforcement rights.

1. Possibility of the issuance of a preliminary injunction without oral hearing

There is a practice in German courts to regularly issue preliminary injunctions without a prior oral hearing. Even though the German Code of Civil Procedure[5] stipulates an oral hearing as the rule[6] and only allows the issuance of a preliminary injunction by court order (an ex parte injunction) as an exception[7] – provided particular urgency is needed – it is nonetheless the case that many regional courts regularly grant preliminary injunctions by court order without hearing the other side in an oral hearing. This practice has been criticised by the German Federal Constitutional Court.[8]

2. Opposition against *ex parte* injunction without suspensive effect

If a preliminary injunction has been issued without an oral hearing, the injunctive relief continues to have effect, at least until a later oral hearing takes place. Thus, opposition alone by the party against whom the injunction has been imposed does not have the effect of suspending the enforcement of the injunction against that party.

3. Possibility of free selection of venue for litigation

As a general rule, in Germany, venue for litigation is given in all places where an infringing act occurs. Thus, if software is offered online for download, the Applicant for injunction is free to choose any venue in Germany where they desire to file a request for the issuance of a preliminary injunction. Thus, the Applicant may request the issuance of a preliminary injunction in any German court, at their discretion, as long as that court is competent to decide questions of copyright.[9] It is obvious that the Applicant will choose a court which has the reputation deciding in favour of the Applicant.[10]

4. Possibility of withdrawing the request for the issue of a preliminary injunction without the consent of the Respondent

Under German law,[11] the Applicant is entitled to withdraw his or her request for the issuance of a preliminary injunction, at any point in time, without the consent of the Respondent. This even applies

5 Zivilprozeßordnung (ZPO).
6 *Id.,* § 128.
7 *Id.*, § 937 (2).
8 German Federal Constitutional Court, Decision of September 30, 2018, File No.: 1 BvR 1783/17.
9 Section 105 of the German Copyright Act authorises the federal state governments to assign, by way of statutory instrument, copyright litigation matters for which the regional court is competent as court of first instance or as appeal court to one of the several regional courts competent within a district.
10 Apparently McHardy considered the first instance court in Cologne, i.e. the Regional Court Cologne, to be Applicant-favourable.
11 ZPO, § 269.

during appeal proceedings.[12] After withdrawal of the request for the issuance of a preliminary injunction in appeal proceedings, the judgment in the first-instance proceedings becomes void. As a result, no judgment of the appellate court is rendered. By this procedural mechanism, the Applicant may attempt to prevent any negative precedent by a higher court.

5. The lack of transparency for German judicial decision-making.

It is German judicial practice to anonymize court decisions.[13] This practice complicates the search for pertinent decisions. Thus, any effort by subsequent Respondents to conduct a particular search for cases where the Applicant has made similar or identical claims is impeded. By allowing an active Applicant to make numerous requests for judicial relief – any of which can be withdrawn for any circumstance, and none of which may be discoverable by subsequent Respondents to understand the nature of prior claims or how they were disposed by the court – encourages a litigation model of pervasive and repetitive requests for injunctions.

6. Requirement of a cease-and-desist declaration subject to a contractual penalty

Pursuant to case law precedent of the German Federal Supreme Court,[14] the danger of recurrence of copyright infringement by a Respondent may only be avoided by having the Respondent provide a cease-and-desist declaration which is subject to a contractual penalty. Respondents are thus put into a bind – either subject themselves to a preliminary injunction, or provide a cease-and-desist declaration subject to a contractual penalty payable to the Applicant – meaning that they must enter into a direct contractual agreement with the person or entity that sent the warning letter. In comparison to an injunction issued by a court, a contractual cease-and-desist declaration is advantageous to the rights owner claiming infringement. In the event of further infringement, the owner of the copyright may claim the contractual penalty payable to that copyright owner, whereas the fines for violation of an injunction are payable to the German state government.[15]

III. Typical tactics in the case of GPL warning letters designed not to ensure compliance, but to collect financial penalty, constitute an abuse of the law

Because of the specific features of German copyright law and procedural practices described above, it can be argued that the tactics used by Applicants sending out GPL warning letters and selectively requesting, and dropping requests for, preliminary injunctions – are abusive of the law. Distributors of code licensed under GPL should be aware of the tactics that are commonly used, and the most effective ways to respond. Typically, these tactics might comprise the following three stages:

1. Requesting an initial cease-and-desist declaration which is subject to a flexible contractual penalty

At the first stage of this type of copyright "trolling," the company receives a warning letter. The letter typically identifies one instance of copyright infringement caused by an inadvertent failure to comply with the basic requirements of GPL – distributing source code, providing a copy of the license, etc. Typically, this instance of infringement is clear and unintentional, and can easily be ceased. Because of this, the company will often not pay particular attention to the request in the warning letter, will

12 Higher Regional Court of Frankfurt, Court Order of March 21, 2018, File No.: 6 W 23/18; Higher Regional Court of Düsseldorf, Court Order of July 13, 1982, File No.: 2 U 54/82.
13 German Federal Supreme Court, Court Order of April 5, 2017, File No.: IV AR(VZ) 2/16.
14 German Federal Supreme Court, Judgment of June 20, 2013, File No.: I ZR 55/12; German Federal Supreme Court, Judgment of July 17, 2008, File No.: I ZR 219/05.
15 Higher Regional Court Cologne, Court Order of May 26, 1986, File No,; 6 W 36/86.

correct the instance of license non-compliance, and will thus sign the cease-and-desist terms, thinking it will end the matter and will be unlikely to result in any further legal difficulties.

In this type of cease-and-desist letter, the company in question is requested to undertake to pay for each and any case of future contravention of the GPL license, and the letter will state that there will be a contractual penalty – the amount of which will be set at the discretion of the person sending the warning letter – and, that in case of a dispute over future violations of the GPL, that dispute will be examined by a court.

Typically, the company is requested to sign a declaration stating that it will cease and desist from making publicly available the Linux kernel and/or distributing the Linux kernel without following the license terms of GPLv2, and that the company will be subject to a contractual fine – as stated above, at the discretion of the copyright holder with whom the cease-and-desist letter is executed – for any case of contravention.[16]

2. Confrontation of the signer of the cease-and-desist letter with a second claim of infringement

The second stage occurs when the company which had received the first warning letter and has signed a cease-and-desist letter is then accused of further infringement. However, in the second instance, the accuser will indicate that signing an additional cease-and-desist declaration subject to a flexible contractual penalty is not sufficient to address the alleged violation. If the company commits another infringement after previously submitting a cease-and-desist declaration subject to a penalty, which penalty is intended to eliminate the danger of recurrence, a new cease-and-desist claim arises against that company.

The new danger of recurrence justified by the renewed allegation of infringement after the submission of a first cease-and-desist declaration can in principle only be eliminated by an additional cease-and-desist declaration with a considerably higher penalty than the first one. The repetition of an identical declaration is claimed to be insufficient. While the addressee of the warning letter need not immediately promise a fixed contractual penalty, or even agree with the accuser's statement about the exact amount of future penalties, case law in Germany requires at least the indication by the accused that they will pay a certain minimum amount – in case where the initial cease-and-desist agreement was subject to a flexible fine.[17] Thus, there might be a tightening of the financial obligation by promising a contractual penalty, inter alia, "*not below* € …." Thus, the accused company, who is already under a contractual cease-and-desist obligation, must now sign another declaration with a penalty clause that states that the penalty amount is now not below a certain amount.

3. Confrontation with a multitude of further infringements and requirement of considerable contractual penalties

The next stage consists of allegations of a multitude of additional claims of infringement, and a demand for payment of considerable contractual penalties. Claims of five-digit amounts (in Euros) per alleged "infringement" are not rare. There have even been reports of considerably higher claims.[18]

Typically, the infringements claimed at this stage are of a different nature than those claimed at the first two stages. In most cases, the company receiving the first warning letter will already have initiated steps to create robust GPL compliance practices, in order to avoid the penalties in the first cease-and-desist declaration. However, in follow-on letters, the infringements the company is

16 According to Section 69c of the German Copyright Act the rights holder, inter alia, has the exclusive right to distribute the computer program and to make it available online.
17 Higher Regional Court of Cologne, Judgement of December 5, 2014, File No.: 6 U 57/14.
18 See, *Edge,* "The rise of copyright trolls," LWN (Linux Weekly News) (May 2, 2017), *retrieved on* Aug. 21, 2018 *under:* https://lwn.net/Articles/721458/ .

accused of at this third stage frequently could comprise rather "exotic" theories of infringement, which might present infringement theories subject to diverging interpretation.[19] However, the company, having initially attempted to accommodate the person sending the initial warning letter, at this point in time, i.e., after having received the second warning letter and having signed the second agreement, is faced with the fact that it had agreed to fixed contractual penalties or a flexible contractual penalty comprising a fixed minimum amount.

In view of the later claimed infringements, the company may realize that compliance with the GPL in all respects, including ways that may not appear consistent with the text of the license or community consensus about license compliance, might be difficult or impossible for legal or technical reasons. The requested compliance might be hard to achieve without substantial resources and a large contingent of experienced experts – although the Linux Foundation offers support in such cases.[20]

At this stage there is a contractual obligation on the accused company to cease and desist from future violations of the license, and that obligation accrues to the benefit of the individual who sent out the warning letter. Thus, the company which has received the warning letters is now under a contractual obligation to cease and desist – to the extent GPLv2 is not completely complied with – and failure to do so subjects them to contractual monetary penalties payable to the original sender of the warning letters. Therefore, the risk that there will be future requests for contractual penalties is high, and more so – with respect to the contractual obligation to cease and desist – it is irrelevant whether the person sending the warning letters actually owns any copyrights in the code for which a future license violation is claimed. In other words, the company is now contractually bound to one individual author to comply with all details of the GPL in future, even for code which that author had no hand in creating. In an instance where there may be violations, high contractual penalties are a significant threat.

If no agreement with respect to the amount of the contractual penalties is reached, the company now faces the risk that a court will order it to cease and desist from further distributing any Linux-based products, including those that might not include copyrighted code from the author with whom the cease-and-desist letter was executed, and that that order might be imposed even for alleged infringements based upon unusual or theories of GPLv2 interpretation that are not generally accepted.

IV. Effective legal arguments that have been used opposing a requested issuance of a preliminary injunction

The Cologne proceedings in the *McHardy v. Geniatech* litigation demonstrate an effective way to oppose a proposed use of a contractual cease-and-desist agreement as a mechanism to extract escalating financial penalties for subsequent allegations of GPL violation. The Applicant, Mr. McHardy, sent a cease-and-desist letter to the Respondent, Geniatech, but Geniatech refused to sign the cease-and-desist declaration. They correctly understood that doing so would subject them to future claims of GPL violations and escalating demands for monetary penalties. In response to the Respondent's refusal to sign the proffered cease-and-desist declaration, the Applicant sent a notification to the Respondent explicitly terminating the Respondent's license to the Applicant's code licensed under GPLv2, and filed an application in the Regional Court of Cologne for a preliminary injunction. The Regional Court of Cologne granted the Applicant's request for an injunction, on an ex-parte basis – without an oral hearing and without even informing the Respondent before issuing

19 The company might, for example, be accused of having not made the offer to make the source code available explicitly valid for at least three years, or that the company did not promptly react to a request for postal delivery of the source code.

20 *See Coughlan/Hemel*, "Practical GPL Compliance," (May 1, 2017), *retrieved on* Aug. 21, 2018 *under:* https://www.linuxfoundation.org/open-source-management/2017/05/practical-gpl-compliance/ .

the injunction.[21] The Regional Court of Cologne even overlooked that the Respondent had filed a protective writ. As mentioned previously (*see* Section II.1 above), German courts often issue ex parte injunctions without an oral hearing or without a chance for the party to which the injunction will apply to be heard. The Respondent filed an opposition to this ex parte injunction and an oral hearing was scheduled, but the Regional Court of Cologne did not re-evaluate the case, instead confirming the injunction.[22] The Respondent filed an appeal to this decision by the Regional Court, and the Higher Regional Court set a hearing for March 7, 2018. In the oral hearing at the Higher Regional Court of Cologne, the Applicant's attorney withdrew his application for injunction.[23]

As a result of the Applicant's withdrawal of its previously-granted injunction request, the Higher Regional Court of Cologne did not render a judgment on the Respondent's appeal. As mentioned previously in Section II. 4 above, in appeal proceedings in Germany, an Applicant can at any point in time simply withdraw their request for relief – even a preliminary injunction that was previously granted – without the consent of the Respondent and with the result that the Court will decline to render a decision for the case on the merits. Thus, by this withdrawal, the Applicant prevented the Court from issuing a decision that may have overturned the previously-granted injunction or would have examined the merits of the underlying claims for which the injunction was granted.

1. No joint authorship exists

The Regional Court of Cologne assumed in its first-instance decision on the Applicant's ex parte request for an injunction that the Applicant was entitled to file the request, and thus entitled to enforce its rights before the Court.[24] By contrast, during the oral hearing of March 7, 2018 at the Higher Regional Court of Cologne, the Court explained its preliminary view that the Applicant's entitlement to file the request had not been shown.[25] In particular, the presiding judge explained in his introductory remarks that in his view the Applicant was not a joint author of the Linux kernel.

On the basis of its preliminary deliberation, the 6th civil division of the Higher Regional Court of Cologne explained that they would deny that the Applicant was a joint author of the Linux kernel. The Court stated that it was its belief that not everyone who had contributed to the Linux kernel could claim co-authorship of the overall program. While the Court reasoned that it was true that for contributions to a piece of software over different stages of time, joint authorship was not generally excluded, such joint authorship within the meaning of Section 8 of the German Copyright Act[26] required that the contributions of the respective contributors could be classified under the common overall idea regarding the work. If later additions and improvements were not directed to the initial programmer's intent to act, no joint authorship was established.[27]

In such a case, the Court reasoned that later changes by the Applicant constituted non-autonomous modifications, if at all. In case of such a modification, the programmer can only raise claims regarding his own contributions provided that these contributions meet the requirements for copyright protection. [28] With respect to the Linux kernel, joint authorship was not given upon initial creation, since the first version had been programmed by Linus Torvalds alone. Indeed, there exist a great number of versions of the Linux kernel which do not include any programming code of the Applicant at all. For example, there exist more than one hundred officially released versions of the

21 Regional Court of Cologne, Court Order of August 23, 2017, File No.: 14 O 188/17.
22 Regional Court of Cologne, Judgment of October 20, 2017, File No.: 14 O 188/17.
23 *See Welte,* "Report from the Geniatech vs. McHardy GPL violation court hearing," (March 7, 2018), *retrieved on* Aug. 21, 2018 *under:* http://laforge.gnumonks.org/blog/20180307-mchardy-gpl ; *Edge,* "A successful defense against a copyright troll," LWN (Linux Weekly News) (April 23, 2018), *retrieved on* Aug 21, 2018 *under:* https://lwn.net/Articles/752485/ .
24 Regional Court of Cologne, Judgment of Oct. 20, 2017, File No.: 14 O 188/17; McHardy/Geniatech.
25 Higher Regional Court of Cologne, Transcript of Oral Hearing of March 7, 2018, File No.: 6 U 162/17; McHardy/Geniatech.
26 German Act on Copyright and Related Rights (Urheberrechtsgesetz, "UrhG"), § 8.
27 German Federal Supreme Court, Judgment of July 14, 1993, File No.: I ZR 47/91.
28 *See* German Federal Supreme Court, Judgment of Mar. 3, 2005, File No.: I ZR 111/02.; Fash 2000.

Linux kernel (version 1.0 – version 2.4.18) completely devoid of any programming code of the Applicant. The list of provisional versions without his contributions is even more extensive.

Pursuant to the reasoning of the Higher Regional Court of Cologne, there was likely only a limited authorship of the Applicant as a result of his contributions, and it is given only to the extent that they constitute real, substantive, contributions and not just editing work or bug fixes. This view of the Higher Regional Court of Cologne is consistent with the license text of GPLv2.

GPLv2 assumes the granting of rights and imposition of obligations as result of the making of modifications. Section 2 of GPLv2 stipulates:

> *"You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of the[] conditions [of the license]."*

The unofficial German translation of GPLv2 uses the term "Bearbeitungen" for the English word "modifications." According to Section 3 of the German Copyright Act "Bearbeitungen" of a work, which are own intellectual creations, are protected as independent works without prejudice to the copyright in the original work. Thus, a person who owns a right according to Section 3 of the German Copyright Act can raise claims regarding his contributions but – in contrast to joint authors[29] – not regarding contributions of the other authors.

The more recent versions of the Linux kernel are based on many prior versions. Thus, typical of open source software, there has been created an extremely long chain of modifications to the first version originally authored by Linus Torvalds. All subsequent versions under German law legally constitute a modification of prior versions.

However, in the *McHardy v. Geniatech* case, with respect to the question of the Applicant's own contributions, the Court indicated that it did not become evident whether the Applicant's contributions are copyright protected at all. It did not even become evident that the Applicant's contributions are protected as modifications within the meaning of Section 3 of the German Copyright Act.

Editing activities or mere bug fixes do not create any authorship on behalf of the individual or individuals engaging in those activities. Instead, Section 69a (3) of the German Copyright Act provides that computer programs shall be protected if they represent individual works in the sense that they are the result of the author's own intellectual creation. Section 69a (3) of the German Copyright Act transposes Art. 1 (3) of the EU Software Directive into German law.[30] Art. 1 (3) of the EU Software Directive provides that a computer program shall be protected if it is original in the sense that it is the author's own intellectual creation. Bug fixes do not meet these requirements.[31] Pursuant to case law,[32] for the establishment of limited authorship in one's own contributions, the Applicant must sufficiently substantiate, and submit evidence as to, the following three factors:

- Which parts of the program have been modified by the alleged copyright author, and in what manner those modifications were made;
- To what extent do those modifications fulfill the requirements for copyright protection; and
- To what extent those modifications by the Applicant were used by the Respondent.[33]

29 Higher Regional Court of Düsseldorf, Judgment of November 25, 2008, File No.: I-20 U 72/06.
30 Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs.
31 *See* Austrian Supreme Court, Judgement of July, 12, 2005; File No.: 4 Ob 45/05d.
32 *See* Regional Court of Hamburg, Judgment of July 8, 2016, File No.: 310 O 89/15, Hellwig/VMware Global, Inc.
33 *Id.*

2. Insufficient substantiation of the protectability of the contributions

In *McHardy v. Geniatech*, the Applicant alleged that over the course of years he had programmed approximately 50,000 lines of code and submitted a CD with changelogs alleged to reflect those changes. However, the Court indicated that this was not sufficient to establish substantiation of the protectability of those changes.[34]

The Court determined that the Applicant did not declare and provide evidence to the Court that he retrieved particular code authored by him in the Respondent's products. The fact that the Applicant was part of the Netfilter Core Team, i.e., the team that was responsible for the Netfilter code in the Linux kernel, did not necessarily mean that he owns any copyrights in that code. Editorial work as such (e.g., bug fixes) does not create copyright protection, because it does not meet the requirements laid down in Section 69a (3) of the German Copyright Act which provides that computer programs shall be protected if they represent individual works in the sense that they are the result of the author's own intellectual creation.

The Court felt that Applicant failed to present and prove, in sufficient detail, in either the initial request for a preliminary injunction or for the submissions on appeal, which parts of the Netfilter program were modified by him and in what manner, to what extent those modifications fulfil the requirements for copyright protection, and to what extent the program parts modified by him had been used by the Respondent. Thus, the question of whether the Respondent used any Linux code did not become relevant.

3. How "Trolling" Activities Can be Seen as an Abuse of the Law

The Higher Regional Court of Cologne did not have to form a final opinion with respect to the objection that Applicant's cease-and-desist activities were an abuse of the law. However, the Court indicated in the oral hearing that an abuse of the law might exist in this case, should it be shown that the pursuit of copyright infringement claims in the Linux kernel formed part of a business model with the aim of achieving profits through compliance errors.

a) Attempt at monetization by means of contractual penalties

An abuse of the law might be found if the enforcement of rights does not aim at GPLv2 compliance, but instead at the achievement of personal financial profits by collecting contractual penalties. This might be found to be the case, e.g., if the business model of the copyright holder provides that the first warning letter does not itemize all known infringements, but the copyright holder only itemizes such infringements after the obligation to pay contractual penalties has already been undertaken. In the initial warning letter, the Applicant requested that various companies sign a broad cease-and-desist declaration as described previously in Section III.1.

b) Termination of GPLv2 by the Applicant

The fact that without prior notice the Applicant terminated the Respondent's rights under GPLv2, pursuant to Section 314 of the German Civil Code, as a reaction to the Respondent's refusal to provide a cease-and-desist declaration subject to a contractual penalty, constitutes further evidence that there was an abuse of the law. While the Regional Court of Cologne found that this termination was inadmissible, the Court did not deduce any abuse of the law.[35]

34 *See* Higher Regional Court of Karlsruhe, Judgment of July 6, 2015, File No.: 6 U 91/15.
35 Regional Court of Cologne, Judgment of Oct. 20, 2017, File No.: 14 O 188/17, McHardy/Geniatech.

However, such termination based on Section 314 German Civil Code is in clear conflict with Section 4 of GPLv2, which stipulates:

> *"You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance."*

German case law interprets Section 4 of GPLv2 in such a way that the license offer made by GPLv2 does not expire definitively in the event of infringements, but instead the infringer can acquire the rights again at any time by accepting and complying with the conditions in the license.[36]

Taking this into account, the Regional Court of Munich has, in *Welte v. Sitecom Deutschland GmbH*, justified the compatibility of Section 4 of GPLv2 with the German Copyright Act. If a termination according to Section 314 of the German Civil Code would actually be permissible, renewed acquisition of the rights provided for by the Regional Court of Munich's *Welte v. Sitecom Deutschland GmbH* decision could be prevented in the case of GPL compliance by the accused infringer. Further, a definitive termination is not in accordance with Section 2 of GPLv2, since every person who modifies the program has to license, as a whole and at no charge, all third parties under the terms of GPLv2, and a definitive termination would take away the rights from the third parties downstream who may not be infringing. The statement of a definitive termination is thus incompatible with GPLv2.

4. The Wording of the request for injunctive relief is too broad

A further essential aspect in the appeal proceedings was that the wording of the request for injunctive relief was too broad.

a) The request for injunctive relief lacked reference to the accused product

The judicial prohibition that the Applicant requested and that the First Instance Court issued was far too broad, since the request did not refer to the specific firmware used by the Respondent Geniatech, but generally only referred to the "Linux kernel." The object of a judgment to cease and desist may only be those acts which have either already taken place, or whose perpetuation is impending.[37] The prohibition must expressly determine which acts must be omitted, and it may not be formulated in such an abstract way that acts might be affected whose lawfulness has not been examined by the Court.[38] Otherwise, the court in charge of execution of the judgment might later have to decide on alleged acts of infringement which do not correspond to the specific acts which were in dispute at the time the injunction was requested. This would mean that the dispute would illegally be transferred to the court in charge of the execution, in case the subject matter has not been examined previously. Therefore, it is up to the court which renders the decision to examine which particular software of a Respondent comprises program lines of an Applicant protected by copyright, and the court may pronounce a prohibition only to this extent.

36 Regional Court Munich, Judgment of May 19, 2004, File No.: 21 O 6123/04, Welte/Sitecom Deutschland GmbH.
37 Section 97 (1) of the German Copyright Act provides: "Any person who infringes copyright or any other right protected under this Act may be required by the injured party to eliminate the infringement or, where there is a risk of repeated infringement, may be required by the injured party to cease and desist. Entitlement to prohibit the infringer from future infringement shall also exist where the risk of infringement exists for the first time."
38 German Federal Supreme Court, Judgement of July 12, 1957, File No.: I ZR 8/56.

b) Unclear reference to the Linux kernel

A request to cease and desist from the distribution of the Linux kernel, or of individual elements of that kernel such as Netfilter, is too vague. This is the case as it remains unclear which version of the Linux kernel is meant when the request for injunctive relief requests only a prohibition on distributing that software. The vagueness is even more decisive if – as in the *McHardy vs. Geniatech* proceedings – it is known that there exist more than one hundred versions of the Linux kernel in which it has become evident that the Applicant did not participate, since they had been finalized and released to the public long before he started working on the kernel.

V. Conclusion

For companies using Linux, cease-and-desist letters claiming insufficient GPLv2 compliance present new challenges. In cases where a demand to agree to broad contractual obligations to cease and desist from using the Linux kernel without complying with the terms of the GPL – combined with contractual penalties – is requested, signing a cease-and-desist declaration is not recommended.

Instead of signing a cease-and-desist declaration, it is advisable that the recipient of the letter initiates two processes immediately. First, the recipient should initiate the technical process of ensuring complete open source license compliance for the software in respect of which a license violation is alleged, as well as beginning to review and correct any other license compliance issues. Second, the recipient should begin preparing to oppose any possible legal action that might follow when the request for a cease-and-desist agreement is rejected, by preparing arguments contesting the asserter's claims of copyright ownership and copyright infringement, and by submitting a protective letter with the court. By doing so, the recipient may be able to force the asserter to prove the substance of their claims and to substantially narrow the scope of any injunction to be issued by the court.

In the former case, the asserter may not be able to establish sufficient basis for an injunction so that the request will be withdrawn or dismissed by the court. In the latter case, the court may issue an injunction that is of sufficiently narrow scope that it will have little negative effect on the recipient's business and may in fact have become moot because of the recipient's efforts to get the product in question into full GPL compliance. This way, the users of GPL software can ensure that efforts to extract escalating penalty revenues by authors of pieces of the Linux kernel ultimately only become a mechanism to get their products into appropriate compliance without negative effect on their product lines, an end goal desired by a large majority of the Linux kernel author community.[39]

About the author

Dr. Marcus von Welser *is a partner in the German IP law firm Vossius & Partner and was admitted to the German Bar in 2002. He advises and represents clients in all areas of intellectual property law, in particular copyright law. He has extensive experience in litigation as well as contract negotiations. The long-standing clients he advises are leading companies from various industries, including software, media and electronics. In 2008, he was awarded the title "Fachanwalt" (certified specialist) in intellectual property law. Since 2009, Dr. von Welser has lectured on international copyright law, enforcement of copyrights, and anti-counterfeiting at Humboldt University, Berlin, Germany. He studied at the universities of Berlin and Warwick (United Kingdom), where he graduated with a*

39 *See* Linux Kernel Enforcement Statement, *reproduced at* https://www.kernel.org/doc/html/v4.17/process/kernel-enforcement-statement.html.

Master of Laws (LL.M.). He further obtained a doctorate degree in copyright law from Christian Albrechts University, Kiel, Germany. Dr. von Welser represented the Respondent, Geniatech, before the Regional Court of Cologne and the Higher Regional Court of Cologne in the McHardy v. Geniatech litigation.

A Survey of Open Processor Core Licensing

Andrew Katz,[a]

(a) Partner, Moorcrofts LLP

DOI: 10.5033/ifosslr.v10i1.130

Abstract
In the Spring of 2018, Western Digital Corporation commissioned Andrew Katz of Moorcrofts LLP to prepare a survey of open processor core licensing. This is an edited version of that report.

Keywords
Law; information technology; Free and Open Source Software; open hardware licensing; processor cores

Foreword by Alan Tse, Western Digital Corporation:

Western Digital's relationship with open source has evolved significantly over the last decade. When I first joined Western Digital, our main focus was on open source compliance. That is because in 2009 we were one of the first major companies sued for our open source use. As a result, the main goal for the next few years was to prevent any litigation happening again and we viewed open source with some trepidation. Over time our view shifted as we had to learn about the importance of the open source community and how to be a good participant in that community. And over the years, we have increased our participation – not to avoid litigation, but because our own business interests have started to align. Over the years we have made multiple contributions to the Linux kernel and other open source projects and we have released internal tools that we thought others could use. Now that it has been almost a decade since that first lawsuit, we would like to think that we have learnt a bit more about the open source community and we are proud to say we are a part of that community.

While we have been public about our support of RISC-V and plans for RISC-V cores since 2017, we also believe the best way to show our commitment to the open source community is by leading from the front. Following our announcement at the December 2018 RISC-V Summit, we recently released our RISC-V SweRV Core under an Apache-2.0 licence on January 24, 2019.[1]

In deciding our licensing strategy for our core release, we engaged Andrew Katz to help us understand the community norms for open source hardware. Owing to his involvement in the drafting of two open source hardware licences, we believed he was at the forefront of Legal scholarship on this issue. His report that follows was instrumental as we balanced our goals of community growth and protection in a space with unique constraints. It's unique because open source

1 For more information see <https://github.com/westerndigitalcorporation/swerv>

hardware is a capital-intensive field quite different from software and the fact that the established open source licences were written without open source hardware in mind. We are happy to release this research to the community and hope this research and our journey serves as a beacon to our peers to join use in the open source hardware community.

– Alan Tse, Associate General Counsel, Western Digital.[2]

Introduction

The research was undertaken by Andrew Katz between March 26 and April 22 2018.[3] The methodology was as follows:

1. To identify major open hardware communities using a combination of research and pre-existing knowledge of various open hardware activities that Andrew Katz has been involved in including both specific projects and umbrella organisations.

2. To undertake research of those organisations and schedule and carry out a range of telephone interviews with identified leading individuals in the field. Given the relatively short time available to undertake the research, a total of eight individuals were identified, of whom six were able to agree to an interview within the time available for the first version of this report. A further two individuals arranged to be interviewed on a date after the original date of submission of the report to Western Digital, and their responses have been taken in to account in the updated version. No one who was approached declined to take part in the research, and all were very open and candid. We are grateful for their time and interest in the project. We also requested further input from the interviewees about community development and involvement, based on the answers to the first round of questions, and two individuals responded comprehensively by email. Their responses were taken into account in the report.

3. To review the projects listed on LibreCores and OpenCores.org, and the list researched by Mohammad Shahrad[4] and updated as a result of further desktop research and responses from interviewees.

4. The results of the research were compiled into this report.

5. In order to facilitate candour on the part of the interviewees, the interviewees were told that their names would not be linked to specific comments they made in a manner similar to the Chatham House Rule. Subsequently, the individuals kindly consented to their names being released.

6. To avoid bias in answers provided, the interviewees were told the research was being undertaken on behalf of a major US digital hardware manufacturer, but no further

2 Alan Tse is a member of Western Digital's Legal team and responsible for open source compliance across the company and supporting Western Digital's open source strategy. His practice covers product lines both up and down the stack including storage devices firmware, consumer devices, data centre systems, and now even hardware cores. As a former computer engineer who grew up using open source software and anxiously waiting for the year of the Linux desktop, he has watched the evolution of open source throughout the tech industry and occasionally dabbles in various open source communities.

3 The data presented in this paper represents information obtained during that research period, unless explicitly stated otherwise. For example, during discussions with Mohammad Shahrad (see footnotes 4, 19 and 20) we agreed to provide him with an updated of the data he presented in the paper referenced at footnote 4, and since this update was provided as at 29 January 2019, we decided to update the relevant text and appendix of this report accordingly. It does not affect the conclusions. The author thanks Heather Stewart for her invaluable assistance in the updating process.

4 Balkind, Joseph, et al. (2016) 'OpenPiton: An Open Source Manycore Research Framework', ASPLOS '16, pp 217 – 232. <http://parallel.princeton.edu/papers/openpiton-asplos16.pdf> DOI: 10.1145/2872362.2872414

information was provided about the research sponsor. The identity of the sponsor was released to the interviewees some months later when they were asked if they were prepared to waive anonymity.

Open Source Hardware and Licensing

Summary

Broad consensus is that 'Open Source Hardware' is hardware whose licensing terms comply with the definition set out by the Open Source Hardware Association. Although the thrust of the definition is relevant to this report, the detail is not.[5] The OSHWA definition follows the Open Source Initiative's definition for Open Source software licensing.[6]

Specific licences which have been identified[7] by OSHWA are:

Copyleft (reciprocal) licences:

- Creative Commons Attribution, Share-Alike (CC-BY-SA)
- GNU General Public License (GPL)
- Hardware-Specific Licenses: TAPR OHL, CERN OHL[8]

Permissive Licences

- Free BSD license (BSD-2-Clause)[9]
- MIT license (MIT)
- Creative Commons Attribution (CC-BY-3.0)[10]
- Hardware-Specific License: Solderpad Hardware Licence[11]

Given that the above licences are specifically referenced by OSHWA we can make the reasonable assumption that they meet the OSHWA definition. OSHWA does not (at the time of writing) have a process for approving licences. It can be assumed that licences (such as Apache) which are approved by the OSI would also meet the OSHWA criteria.

Licences that were identified during the course of this survey as applying to various open source hardware projects are:

5 For more information see <https://www.oshwa.org/definition/>
6 With the interesting distinction that in the preamble, OSHWA states that the design must be publicly available so that anyone can make etc. the design. OSI only requires that the licensing terms enable the licensee to make open source software publicly available, but not that public availability itself is necessary.
7 <https://www.oshwa.org/sharing-best-practices/>
8 Andrew Katz has been involved in the drafting of CERN OHL <https://www.ohwr.org/documents/294>.
9 <https://opensource.org/licenses/BSD-2-Clause>
10 <https://creativecommons.org/licenses/by/3.0/>
11 Andrew Katz drafted the Solderpad Licence. <http://solderpad.org/licenses/>

Licence	Comments
BSD-2-Clause (simple permissive)	Widely used for many types of open source hardware, including processor cores
MIT (simple permissive)	Widely used for open source hardware
ISC[12] (simple permissive)	Sometimes used for open source hardware
Apache-2.0 (permissive with patent clauses)	Widely used for open source hardware
GPLv3 (strong copyleft with patent licence)	Frequently used for open source hardware
GPLv2 (strong copyleft without patent licence)	Frequently used for open source hardware
LGPL (various versions)	Frequently used for open source hardware
MPL-2.0 (weak copyleft with patent grant)	Rarely used for open source hardware

Table 1: Open Source Software Licences

Licence	Comments
Creative Commons Attribution (various versions)	Widely used for open hardware designs
Creative Commons Share-Alike (various versions)	Widely used for open hardware designs
Creative Commons Public Domain Dedication (CC0)	Widely used for open hardware designs

Table 2: Open Content Licences

Licence	Comments
TAPR (Tucson Amateur Packet Radio) Open Hardware License	Mainly used for RF circuit boards. Has interesting copyleft mechanism, based on patents
CERN OHL (various versions)	Used for a wide variety of open hardware but originally designed mainly for applicability to circuit boards
Solderpad Licence (Versions 0.51 and 2) (an Apache-based Open Hardware License)	Used for a wide variety of hardware, including cores
Open Hardware Description Licence (Mozilla Public License-based open software licence)	Designed specifically for semiconductor cores. Rarely used.
NVDIA Open NVDLA License and Agreement v1 (an Apache-based Open Hardware License)	Designed specifically for NVDLA (Nvidia Deep Learning Accelerator)

Table 3: Hardware Specific Licences

12 <https://opensource.org/licenses/ISC>

Licence	Comments
Public Domain Dedication	Public domain dedication is not recognised in many jurisdictions, although it may take effect as a broad licence. CC-0 (see above) seeks to remedy this by providing an explicit fallback licence
Creative Commons NC variants	Non-commercial licences contain a restriction against a field of endeavour (commerce) contrary to paragraph 8 of the OSHWA criteria
Open Compute Project Licences (passive and copyleft)	Designed for hardware for use in OCP-compatible databases. The licences only really work when the various participants are patent holders, and are better regarded as standards-coupled licences

Table 4: Licences which are not compliant with OSHWA/ODI criteria

Note that both the Solderpad licence and the CERN OHL are in the process of revision. Version 2 of the Solderpad licence remains very similar to the Apache licence it is based on, but has been amended so that is now expressed to be a 'wraparound' of the Apache licence, rather than expressed as a different license. The advantages are that it is much easier for a practitioner familiar with Apache 2.0 to immediately see what the differences are between Solderpad and Apache 2.0.

As of January 2019, the CERN OHL is in the process of being modified significantly to produce version 2. Under current proposals this will be published in three variants: a permissive version which has an Apache-like effect, and two reciprocal versions – lesser and strong (strong reciprocal being the default for those who have already published hardware designs under current versions of the CERN-OHL with the ability to select a later version). Care has been taken to consult with developers of FPGAs and ASICs to try to meet their concerns, particularly around the use of proprietary tools and libraries that are all but unavoidable in practice, while retaining the copyleft nature of the reciprocal versions of the licence.

Desktop Analysis of Licence Adoption

The OSHWA Surveys

Across open hardware as a whole probably the most in-depth survey of open source hardware use and attitudes was undertaken by the OSHWA in 2012[13] and 2013.[14] This contained a small section on licence adoption. It should be noted that this survey covered open hardware in general, from mechanical items through to electronics, but there is no indication that any of those responding were involved in development at sub-component (i.e. chip design) level. Therefore, the results are both fairly out of date and of dubious relevance to chip design. One section of the survey related to licence adoption, and like the annual Black Duck licence adoption survey[15] counts all projects of equal weight. For example, in the Black Duck survey the Linux Kernel counts as a project with equal weighting to a tiny driver project which appears on GitHub but has never been used in commercial deployment). The results are therefore a dubious reflection of reality though it is interesting to note that very nearly 50% of the respondents had released projects with no explicit licence. It is difficult to interpret the results, as each respondent was permitted to respond with multiple answers to the

13 <https://www.oshwa.org/oshw-community-survey-2012/>
14 <https://www.oshwa.org/oshw-community-survey-2013/>
15 <https://www.blackducksoftware.com/top-open-source-licenses>

question of which licence they had used, but the thrust for open hardware in general (covering everything from mechanical devices and casings through to circuit boards) is that there is rough equality of deployment of copyleft licences (e.g. Creative Commons Share-Alike, GPL) and permissive (e.g. MIT, BSD, Creative Commons attribution-only) licences.

GitHub Search

Many open hardware projects are hosted on GitHub. CERN carried out some basic research on how many projects have adopted the CERN OHL by carrying out a Google search for "site:github.com CERN-OHL" that as of March 2018 produced[16] 657 results. It is misleading to assume these are all projects. However, undertaking a random sample of 10 pages from the complete Google results shows that around 80% of the results **are** projects. It is not easy to tell if these are unique results, but if they are, it suggests that something over 500 CERN-OHL licensed projects exist on GitHub. By comparison, TAPR OHL only generates 39 results of which 15 appear to be projects.[17] Solderpad shows 434,[18] almost all of which appear to be legitimate projects. It should be noted that it is more difficult to use this sort of search to find hardware projects licensed under Apache, MIT or BSD for the simple reason that the search will generate, overwhelmingly, software projects.

The OpenPiton Survey[19]

As part of a 2016 paper, Mohammead Shahrad, a member of the Princeton OpenPiton team, researched active processor core projects.[20] We have updated, corrected and verified the information presented and a summary in the table in appendix 2 under the section 'OpenPiton'.[21]

Of particular interest is that, when the projects are listed in order of the date of last active contribution, it is clear that the more recent projects are more heavily weighted towards permissive, rather than copyleft licensing. There is a total of 28 processor core products listed. There is a gap between October 2015 and February 2017, and if we take the projects that have been active since February 2017 (of which there are 15), 5 of them are copyleft. For projects prior to this date (of which there are 13), 12 are copyleft.

To summarise: recently active projects are split 33% copyleft, 67% permissive, as against the non-active projects, which are 92% copyleft, 8% permissive. This indicates a clear shift to permissive licensing for currently active projects.

16 As of 29 January 2019, this has increased to 'about 1500', but the search results are somewhat noisier, so it's not clear if this is a valid comparison.

17 We tried to rerun the search on 29 January, but the results were so much noisier that it's impossible to make a valid comparison.

18 We tried to rerun the search on 29 January, but the results were so much noisier that it's impossible to make a valid comparison.

19 Balkind, Joseph, et al. (2016) 'OpenPiton: An Open Source Manycore Research Framework', ASPLOS '16, pp 217 – 232. <http://parallel.princeton.edu/papers/openpiton-asplos16.pdf> DOI: 10.1145/2872362.2872414

20 <http://parallel.princeton.edu/openpiton/open_source_processors.php>

21 The results in the appendix have been updated to 29 January 2019, and therefore differ slightly from the version of the table provided to WD in the original version of the report. The figures above have been updated accordingly. For comparison, the text in the original report read: "There is a gap between October 2015 and February 2017, and if we take the projects that have been active since February 2017 (of which there are 12), 5 of them are copyleft. For projects prior to this date (of which there are 14), 12 are copyleft."

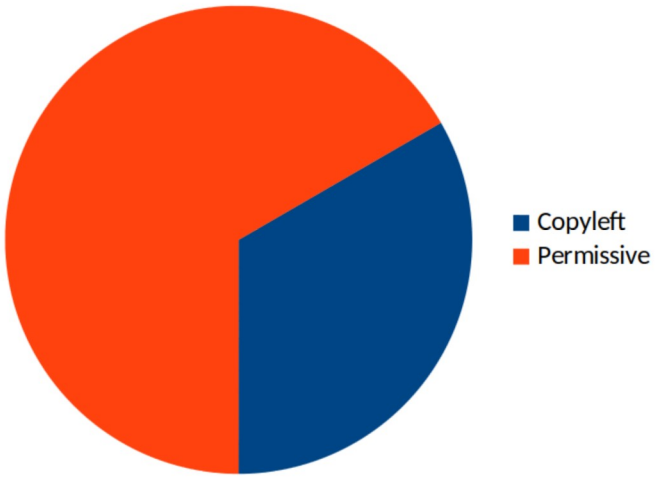

Fig. 1: Summary of licences chosen for recently active projects, data from the OpenPiton Survey

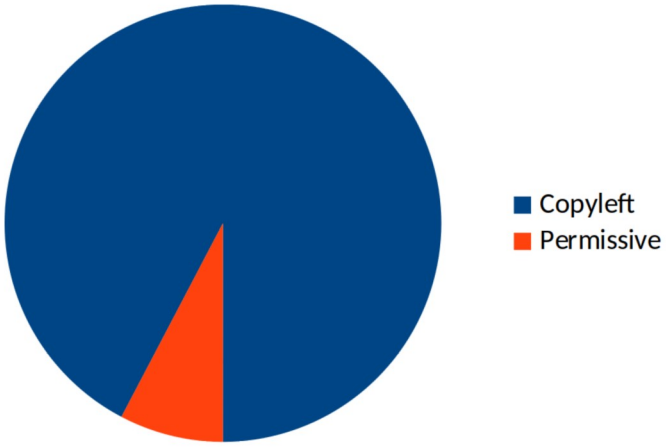

Fig. 2: Summary of licences chosen for non-active projects, data from the OpenPiton Survey

OpenCores and LibreCores

Two websites, opencores.org and librecores.org, host core designs and related materials such as tools and interfaces ('interfaces' are materials for other components which would typically appear on silicon alongside a core, such as UARTs and memory controllers). Opencores is run by a commercial entity, a situation which led to dissatisfaction from members of the FOSSi foundation regarding how Opencores operated, and their subsequent creation of Librecores as an alternative. Librecores has fewer projects, but they tend to be more active than Opencores (possibly because they have had less time to become obsolete).

There is a total of 1190 entries on the Opencores website, including software, toolchains, utilities and interfaces, as well as cores, of which 30 are marked verified. The Librecores site contains 90 entries but does not have any form of verification mechanism. We examined 24 entries in the Opencores website which are marked as 'verified' and 40 entries on Librecores. We selected entries which most

clearly relate directly to cores and interfaces, details of which are contained in appendices 3 and 4. There is also a thriving ecosystem of associated software tools, test suites and build and utility scripts, analysis of which is outside the scope of this report. Whilst we have undertaken a statistical analysis of this data, it is important to note that should be interpreted in the light of the following constraints:

(1) there is no easy way to weight each entry in terms of how pervasive and active the project is, so a barely-functional and rarely-adopted project would rank the same as a more mature and active one;

(2) there are significant projects which are not represented on either database;

(3) the selection of entries is largely subjective, and whilst the intention is to select projects which instantiate hardware (as opposed to toolchain or utility components), the selection was undertaken by a lawyer and not a microelectronics engineer, so mistakes are inevitable.

Various analyses of the licensing in both Opencores and Librecores for various categories of project are set out on Appendices 3 and 4.

Outcomes of the telephone interviews

Licensing – copyleft vs. permissive

All but one interviewee noted that a permissive model was the most likely to succeed from a commercial perspective. All acknowledged that a particular issue with copyleft licensing was that existing licences, including GPL and LGPL, and even CERN OHL did not provide sufficient certainty as regards boundaries delineating where the copyleft effect occurs. For example, if a component whose design is released under LGPL is combined with another component on the same silicon, does that mean that both components then have to be released under the LGPL? How about if the components are on separate chips? One interviewee specifically referred to the little-understood requirement in LGPL for sufficient interface information to be made available (together with the right to reverse engineer), for the LGPL component to be modified and re-linked to the 'work' as a whole. It is not clear how that would work with electronics especially since the works could be combined on static silicon (as masks). One interviewee noted that OpenSPARC (which was licensed under GPLv2) had in the past proved to a successful design (used for devices as diverse as digital cameras and network interfaces), thus demonstrating that GPL-based designs are capable of being commercially successful. There is little publicly available information on OpenSPARC (which is a relatively old project, having been released in 2006), and the interviewee suggested that separate research should be undertaken by locating some of the individuals who had been involved in the project initially, and in particular, the decision to open the technology, and to interview them.

Horizontal and Vertical Boundaries

Another interviewee made the explicit distinction between 'horizontal' boundary problems (as mentioned above), and 'vertical' boundary problems where it is not clear whether a requirement to release design documentation for a circuit design (or similar) also requires releasing the designs of the components themselves. It was noted that the CERN OHL explicitly deals with this via the requirement to release information for modifications at a similar 'level of abstraction' to the original design.[22] The current version of CERN OHL does not deal with the horizontal boundary problem

22 At of January 2019, proposals for CERN OHL v2 take a different approach and have introduced the concept of an 'Available Component'. Designs do not have to provide the design documentation for components that qualify as 'Available Components', which include items like readily available electronic components, provided that enough information about their specification, characteristics and interfaces is available to enable them to be sourced or used in the design. Thus a 555 timer when provided along with its datasheet would quality as an 'Available Component'.

(although this is to be addressed in the upcoming version 2 as mentioned above). The Open Hardware Description Licence (based on Mozilla Public License 2.0) does address this problem but is not frequently used.

One interviewee suggested the horizontal boundary problem might be fixed by saying that a weak copyleft licence could be drafted in a manner that the licensor provided an interface definition alongside the code. Provided that any third party complied with the interface definition, their code linking to the original licensor's code would be free of the reciprocal effect. The next version of the CERN OHL – see above – is likely to adopt an optional mechanism similar to this.

What drives licence choice (copyleft vs. permissive)?

Most interviewees expressed a preference for permissive licensing on the basis that existing copyleft licences left too much uncertainty, and that this uncertainty would inhibit adoption. It would also make it more difficult to deal with companies which provide proprietary libraries as those companies would be uncomfortable having their proprietary library used in a design which covered by copyleft of uncertain scope. One interviewee noted the value in copyleft licensing and noted that the Open Hardware Description licence expressly addressed the scope problems, but that it had not been widely adopted.

When it was suggested in each case that the next version of the CERN OHL would likely incorporate additional optional exceptions which expressly limited the reciprocal effect (as noted above) respondents suggested that this would cause them to potentially reconsider their licensing choices and consider its adoption. However, that there was little point in examining the issue in greater depth until such a licence was more widely accepted in the wild.

It was generally accepted that licence choice was ideological, and that some projects would be more inclined to wish to maximise use of their designs by providing them under a permissive licence while accepting the danger that the designs may become incorporated into proprietary hardware, while others wished to maximise freedom by making them available under a copyleft licence which ensured modifications would be made available under the same licence. However, all parties were all uncomfortable with existing copyleft licences, and regarded the issue, as this stage, as being largely hypothetical.

One interviewee noted that the use of components under copyleft licences in their current state would potentially cause difficulties with fundraising. One interviewee noted that in a due diligence exercise it was not unusual to run a code-scanning tool such as Black Duck against HDL files, although it is not immediately clear what the benefits of such an activity would be and whether Black Duck holds any HDL in its codebase, other than potentially to scan the code for licence texts such as the GPL which are frequently regarded as 'risky' by funders.

'Selling exceptions to the GPL'

One interviewee did note that it was possible that a design could be licensed under a restrictive copyleft licence of uncertain scope with respect to hardware such as the GPLv2 with a view to the licensor making a parallel proprietary licence providing certainty available for a fee. Clearly, this model tends to cause the licensor to use more restrictive licences in an effort to drive adoptees to the proprietary licence, whilst still permitting the licensor to describe the designs as 'Open Source [hardware]'. Richard Stallman, founder of the Free Software Foundation, has described this practice disparagingly as 'selling exceptions to the GPL'.[23]

23 Stallman, Richard 'Selling Exceptions to the GNU GPL' <https://www.gnu.org/philosophy/selling-exceptions.en.html>

One interviewee provided, as an example, the Leon core provided by Gaisler and based on SuperSPARC that is available both under LGPL/GPL and a proprietary licence. This was simply an illustration of dual licensing and does not suggest any particular motivation on the part of Gaisler for choosing that licensing model.

Open Hardware Communities

The consensus among interviewees was that the lack of open source or low-cost toolchains was an inhibiting factor in the growth of open hardware communities focusing on cores.

It is noteworthy that cores which emulate obsolete or obsolescent designs, primarily of interest to hobbyists, are more likely to be licensed under copyleft licences. For example, the Neo430, OpenMSP430 and T400 and T48 μController cores, examples of cores from selected OpenCores projects which fall into this category, are all licensed under copyleft licences.

After the initial phase of interviews, a second set of questions were sent to the interviewees focusing specifically on community building. We received two comprehensive responses within the time available, and both noted that permissive licences would be more attractive to commercial projects owing to avoidance of the problems around perceived linking. Both also pointed out that there probably was not enough data available to determine whether projects using non-open-hardware licences would have chosen an open hardware-specific licence if one was available. A potential illustration of this is that the OpenPiton list only three projects out of 26 chose a hardware specific licence (in all cases, the Solderpad licence.[24] In no case was a hardware-specific copyleft licence chosen.

Both responses also indicated that, most commonly, projects based on a permissive licence retained the same licence when out- bound licensing (i.e. the licence under which the design is to be licensed to third parties), as for the in-bound contributions.

In terms of community development, interviewees stressed the importance of evangelism and outreach, and funding community development. One individual also stressed the importance of becoming involved in projects like the FOSSi foundation.

Toolchains

One issue that came up frequently, although detailed discussion is outside the scope of this report, was that open source toolchains are much scarcer in the world of open hardware than they are in software. The extent to which the toolchain will incorporate code of its own into the output, and what the effect of that code is from a legal point of view, is highly problematic: it is a debatable point as regards software but becomes even more so when applied to hardware. Questions arise such as whether a bitstream is in any sense a computer program, and - if so - who 'runs' it when the hardware starts up.

Patents

The interviewees generally noted that patents were a potential problem but had no clear suggestions on how to address this challenge. It was noted that members of the RISC-V foundation get the benefit of a cross-licence agreement from the other members, but that non-members, although they are able to use the ISA specification freely, gain no form of explicit patent licence or protection.

24 The results in the appendix have been updated to 29 January 2019, and are not the version of the table provided to WD. The text of the version of the report released to WD accordingly read "two projects out of 26" in the sentence to which this is a footnote.

One interviewee noted that there was a move towards licences such as Apache 2.0 away from BSD or MIT because of its explicit patent provisions. One noted that the Solderpad licence (an Apache variant) had been adopted by LowRISC and PULPino because it was a relatively simple licence which had been modified specifically for hardware and had Apache-like patent provisions.

Establishing a default licence to use - recommendations

Broadly, licence choice should be limited to one of the more popular licences. Which specific licence is chosen depends depending on business needs for that the relevant project. The most popular software licence choices include the licences of the GPL family, Apache 2.0 and potentially MPL. For hardware, these may roughly correlate with CERN OHL/TAPR, Solderpad or BSD/MIT and Open Hardware Description License. Less well-used licences should be avoided, because they may cause licence incompatibility problems, and it makes project adoption more problematic. It is worth bearing in mind that the lawyers acting for counterparties prefer to work with the text of better-known licences to avoid having to spend expensive time to become familiarised with them. The informal drive towards licence standardisation is a topic which arises at legal licensing conferences quite frequently: the observation goes that the GPL, for all its flaws, is well understood, so tends to lead to a better legal outcome for both parties in contrast to a licence like (for example) the Open Software Licence, which is arguably better drafted, but less well used and understood.

It may be the case that there is, in practice, no choice that can be made, if the project uses, for example, a GPL component at its core which cannot be sufficiently decoupled from the rest of the work. In that case, the whole project would likely have to be released under the relevant version of the GPL.

For projects where there is no such constraint the specific choice of licence will depend upon the criteria of the specific project. The key question is whether the licensor is seeking to maximise either *utilisation* or *freedom*.[25]

If the licensor is seeking to maximise utilisation then a permissive licence such as Solderpad[26] will be most appropriate. In this case, the licensor must be comfortable that the software or hardware design may be incorporated into proprietary systems, and that the source code/design of any modifications may not be made available.

On the other hand, to maximise freedom, a good choice is the CERN OHL (adopting, where appropriate, one of the reciprocal versions, when v2 is released).

Another option as referred to previously in this paper is to sell proprietary unrestricted licences alongside a given open source licence (assuming the licences of the other components allow this). It is common practice to use a restricted licence (such as GPL or CERN OHL with no exceptions) to enhance the attractiveness of the proprietary option, though while this is common it is frowned upon by the GPL community. On the other hand, a legitimate reason for dual licensing may be that the licensee wishes to use a GPL-licensed core alongside third-party proprietary components, and therefore has to seek a licence from the licensor of the core which is compatible with those components.

25 In the sense of 'liberty'. In other words, the designer's intention is that the design, in all its incarnations, remains free of constraints on reuse, modification and distribution, and also has the effect of causing other designs combined with it to be equally free, with the overall intention of increasing the commons of free designs.

26 Or the newly (January 2019) announced permissive version of the CERN OHL.

Conclusion

All interviewees believed that the most commercially effective open hardware core designs were those which adopted permissive licences. The prevalence of these licences is borne out by desktop research. The stated various reasons for this are:

- that the currently available copyleft open hardware licences are insufficiently clear in their effect to be safely used;
- that the potential benefits of copyleft licensing in core designs are not yet sufficiently clear to show an overwhelming need to shift to a copyleft model;
- that copyleft licensing is certainly interesting and may have a place as the market matures. No interviewee was against copyleft core licensing in principle (although there was consensus that a weak copyleft with clearly defined boundaries was more likely to be commercially successful).[27]

Note that even though the interviewees selected were intended to represent a cross section of the core-developing communities, RISC-V was referred to by every interviewee. The emphasis on permissive licensing may therefore be an artefact of the relatively small sample size and a shared familiarity by the interviewees with RISC-V. It may, on the other hand, reflect a reality that RISC-V is the most prominent and widely adopted open ISA currently in use.

About the author

Andrew Katz *is a partner at boutique law firm Moorcrofts LLP, based in England's Thames Valley. Andrew specialises in technology law and has a particular interest in open design and development. He can be contacted at andrew.katz@moorcrofts.com*

27 At the time of the original research, some of the interviewees were aware that CERN was in the process of redrafting the CERN OHL to create a new version intended to address the concerns of FPGA and ASIC developers. The approach which was being taken at that stage was by way of application-specific exceptions to the licence. The current approach (as at January 2019) is somewhat different and now allows code libraries, macros, etc. which are provided as part of the toolchain to be included as an 'Available Component' - see footnote 22.

Appendix 1

Interviewees

- Krste Asanovic, Dept EECS, UC Berkeley.
- Andrew Back, Managing Director, AB Open.
- Julius Baxter, Director, FOSSi Foundation.
- Dr Jeremy Bennett, Chief Executive, Embecosm.
- Alex Bradbury, lowRISC CIC.
- David May, Professor of Computer Science, Bristol, Founder XMOS, FREng, FRS.
- Simon Phipps, Founder, Meshed Insights Ltd.
- Dr Davide Rossi, University of Bologna.

Appendix 2

Taxonomy of Open Source Processors from OpenPiton

Processor	Architecture	Licence	Last Update to Project	Last Update to Code
aeMB	32b MicroBlaze	LGPL v3	Feb 2012	-
AltOr32	32b ORBIS	LGPL v3	Feb 2015	Jun 2014
Amber	32b ARM v2a	LGPL	Sept 2017	Nov 2015
Ariane	64b RISC-V	Solderpad	Jan 2019	-
BERI	64b MIPS/CHERI	BERI HW-SW	Mar 2017	-
CPU86	16b x86	GPL	Jun 2014	-
LatticeMicro32	32b LatticeMicro32	GPL	Oct 2017	-
LEON 3	32b SPARC v8	GPL	Dec 2017	-
MIAOW GPGPU	AMD Southern Islands	BSD 3-Clause & GPL v2	Sept 2017	-
MIPS32 r1	32b MIPS 32 rl	LGPL v3	Jul 2015	-
mor1kx	32b ORBIS	OHDL	Jan 2019	Jan 2019

Processor	Architecture	Licence	Last Update to Project	Last Update to Code
openMSP430	16bMSP430	BSD	May 2018	Apr 2018
OpenPiton	64b SPARC v9	BSD 3 Clause & MIT	Jan 2019	-
OpenRISC	32b/64b ORBIS	LGPL	Nov 2018	-
OpenScale	32b MicroBlaze	GPL v3	Jan 2012	-
OpenSPARC T1/ T2	64b SPARC v9	GPL v2	Nov 2008	-
or1200	32b ORBIS	LGPL	Oct 2015	Jun 2015
pAVR	8b AVR	GPL v2	Jul 2009	Mar 2009
Pico RV	32b RISC-V	ISC	Nov 2018	Nov 2018
PULP-RI5CY	32b RISC-V	Solderpad	Jan 2019	-
RISC-V Boom	64b scalar RISC-V	BSD 3-clause	Jan 2019	-
RISC-V Rocket	64b scalar RISC-V	BSD 3-clause	Jan 2019	-
SecretBlaze	32b MicroBlaze	GPL v3	Dec 2012	Dec 2012
Simply RISC S1	64b SPARC V9	GPL v2	Dec 2008	-
XUM	32b MIPS32 r2	LGPL v3	Jul 2015	-
Zeroriscy	32b RISC-V	Solderpad	Nov 2018	-
Zet	16b x86	GPL v3	Nov 2013	-
ZPU	32b MIPS	FreeBSD + GPL	Apr 2015	-

Table 5: Taxonomy of differences of open source processors (table data last checked 29 January 2019). [28]

28 Originally published in Balkind, Joseph, et al. (2016) 'OpenPiton: An Open Source Manycore Research Framework', ASPLOS '16, pp 217 – 232. <http://parallel.princeton.edu/papers/openpiton-asplos16.pdf> DOI: 10.1145/2872362.2872414, Table 4, updated at http://parallel.princeton.edu/openpiton/open_source_processors.php

Appendix 3

OpenCores

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
Elliptic Curve Group (ecg)	OpenCores	The Elliptic Curve Group core is for computing the addition of two elements in the elliptic curve group, and the addition of c identical elements in the elliptic curve group and it is carefully optimized for FPGA	LGPL	Component	Copyleft
Reed Solomon Decoder (204,188)	OpenCores	Reed Solomon Decoder (204,188), with T=8	GPL	Component	Copyleft
Viterbi Decoder (AXI4-Stream compliant)	OpenCores	A fully configurable VHDL Viterbi decoder compliant with the AXI4-Stream interface	GPL	Component	Copyleft
Ethernet 10GE MAC (xge_mac)	OpenCores - GitHub	The 10GE MAC Core implements the Media Access Control functions for 10Gbps operation as defined in IEEE Std 802.3ae.	LGPL	Interface	Copyleft
Ethernet MAC 10/100 Mbps (ethmac)	OpenCores	The Ethernet MAC (Media Access Control), sublevel within the Data Link Layer of the OSI reference model. This core is designed for implementation of CSMA/ CD LAN in accordance with the IEEE 802.3 standards.	LGPL	Interface	Copyleft
sd card controller (sdcard_mass_storage_controller)	OpenCores	The "sd card controller" is a Secure Digital Card Host Controller, which main focus is to provide fast and simple interface to SD/SDHC cards.	LGPL	Interface	Copyleft

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
Small 1-wire (onewire) master, with Altera tools integration (sockit_owm)	OpenCores	This IP implements the 1-wire communication protocol (http://en.wikipedia.org/wiki/1-Wire).	LGPL	Interface	Copyleft
PCIe SG DMA controller	OpenCores	This package involves a PCIe Scatter-Gather DMA engine for Virtex5 and Virtex6 and implements MAC, Physical (Xilinx Hard and Soft IP Cores) and Transaction Layer (Custom Core) of PCIe.	LGPL	Interface	Copyleft
Wupper: PCIe DMA Engine for Xilinx FPGAs (virtex7_pcie_dma)	OpenCores	A system controller primarily designed to provide an interface to standard FIFOs (a simple Direct Memory Access (DMA) interface to the Xilinx Virtex-7 PCIe Gen3 hard block.)	LGPL	Interface	Copyleft
8/16/32 bit SDRAM Controller (sdr_ctrl)	OpenCores - GitHub	8/16/32 Configurable SDRAM data width which is Wish Bone compatible.	GPL	Interface	Copyleft
High Performance Dynamic Memory Controller (hpdmc)	OpenCores	HPDMC is part of the Milkymist System-on-Chip, the most advanced open source SoC for interactive multimedia applications.	GPL	Interface	Copyleft
VGA/LCD Controller (vga_lcd)	OpenCores	The OpenCores VGA/LCD Controller core is a WISHBONE revB.3 compliant embedded VGA core capable of driving CRT and LCD displays. It supports user programmable resolutions and video timings, which are limited only by the available WISHBONE bandwidth.	GPL	Interface	Copyleft

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
I2C controller core (i2c)	OpenCores - GitHub	I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange between devices. It is primarily used in the consumer and telecom market sector.	BSD	Interface	Permissive
UART to Bus (uart2bus)	OpenCores	The UART to Bus IP Core is a simple command parser that can be used to access an internal bus via a UART interface and provides a quick and easy way to test a new FPGA board.	BSD	Interface	Permissive
Plasma - most MIPS I(TM) opcodes (plasma)	OpenCores	The Plasma CPU is a small synthesizable 32-bit RISC microprocessor currently running a live web server with an interrupt controller, UART, SRAM or DDR SDRAM controller, and Ethernet controller.	Others	Pcore	
Tate Bilinear Pairing	OpenCores	The Tate Bilinear Pairing core is specially designed for running Tate bilinear pairing algorithm for hyperelliptic curve $y^2=x^3-x+1$ defined over $GF(3^m)$, where $m=97$ and $GF(3^m)$ is defined by x^97+x^12+2 and it is carefully optimized for FPGA.	LGPL	Pcore	Copyleft
Amber ARM-compatible core (amber)	OpenCores	The Amber processor core is an ARM-compatible 32-bit RISC processor. The Amber core is fully compatible with the ARM® v2a instruction set architecture (ISA) and is therefore supported by the GNU toolset.	LGPL	Pcore	Copyleft
NEO430 Processor (MSP430-compatible)	OpenCores and librecores	This processor is based on the Texas Instruments MSP430 ISA and provides 100% compatibility with the original instruction set but is not an MSP430 clone.	LGPL	Pcore	Copyleft

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
minsoc	OpenCores	The Minimal OpenRISC System on Chip is a system on chip (SoC) implementation with standard IP cores available at OpenCores.	LGPL	Pcore	Copyleft
CORDIC core	OpenCores	The CORDIC algorithm is an iterative algorithm to evaluate many mathematical functions, such as trigonometrically functions, hyperbolic functions and planar rotations.	GPL	Pcore	Copyleft
T400 µController (t400)	OpenCores	The T400 µController is an implementation of National's 4-bit COP400 microcontroller family architecture intended to be used as a replacement for the original chip in SOCs recreating legacy systems.	GPL	Pcore	Copyleft
T48 µController	OpenCores	The T48 µController core is an implementation of the MCS-48 microcontroller family architecture. While being a controller core for SoC, it also aims for code-compatability and cycle-accuracy so that it can be used as a drop-in replacement for any MCS-48 controller.	GPL	Pcore	Copyleft
openMSP430	OpenCores - librecores	The openMSP430 is a synthesizable 16bit microcontroller core written in Verilog. It is compatible with Texas Instruments' MSP430 microcontroller family and can execute the code generated by any MSP430 toolchain in a near cycle accurate way.	BSD	pcore	Permissive

Table 6: OpenCores

Appendix 4

Librecores

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
ZAP ARM Processor	librecores	ZAP is a pipelined ARM processor core that can execute the ARMv4T instruction set. It is equipped with ARMv4 compatible split writeback caches and memory management capabilities.	GPL	pcore	Copyleft
mor1kx	librecores	This repository contains an OpenRISC 1000 compliant processor IP core.	MPL 2.0 RC2	pcore	Copyleft
neo430	librecores	This processor is based on the Texas Instruments MSP430 ISA and provides 100% compatibility with the original instruction set but is not an MSP430 clone	LGPL	pcore	Copyleft
kpu-soc	librecores	KPU is a minimal system on chip (SoC) created for use as a testbench for the KPU core	GPL	pcore	Copyleft
PULPino	librecores	Single-core microcontroller system based on 32-Bit RISC-V cores (ETH Zurich)	SOLDERPAD HW LICENCE V0.51	pcore	Permissive
parallella-riscv	librecores	Integration of the RISC-V rocket core, inside the Zynq FPGA device of Parallella	MIT and The Regents of the University of California	pcore	Permissive
RgGen	librecores	Code generation tool for control/status in a SoC design	MIT	pcore	Permissive
picorv32	librecores	PicoRV32 is a CPU core that implements the RISC-V RV32IMC Instruction Set. It can be configured as RV32E, RV32I, RV32IC, RV32IM, or RV32IMC core, and optionally contains a built-in interrupt controller.	ISC	pcore	Permissive

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
SimpleVOut	librecores	A simple set of FPGA cores for creating video signals in various formats.	ISC	pcore	Permissive
NyuziProcessor	librecores	Nyuzi is an experimental GPGPU processor hardware design focused on compute intensive tasks. It is optimized for use cases like deep learning and image processing.	Apache v2.0	pcore	Permissive
riscv-sodor	librecores	educational microarchitectures for risc-v isa	Sodor based on the BSD 3-clause licence	pcore	Permissive
TV80 Z80-compatible microprocessor	librecores	TV80 is a Z80-compatible synthesizable Verilog core and aims to be an area-efficient core which closely mimics the original operation and cycle timing of the Zilog Z80.	MIT	pcore	Permissive
Ariane	librecores	Ariane is a 6-stage, single issue, in-order CPU which implements the 64-bit RISC-V instruction set. It has configurable size, separate TLBs, a hardware PTW and branch-prediction (branch target buffer and branch history table). The primary design goal was on reducing critical path length.	Solderpad v0.51	pcore	Permissive
RV12 RISC-V Processor	librecores	The RV12 is a highly configurable single-issue, single-core RV32I, RV64I compliant RISC CPU intended for the embedded market.	other	pcore	
openGFX430	librecores	The openGFX430 is a synthesizable Graphic controller written in Verilog and tailored for the openMSP430 core.	3-Clause BSD	interface	Permissive

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
liteeth	librecores	LiteEth provides a small footprint and configurable Ethernet core whose aim is to lower entry level of complex FPGA cores used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller.	2-clause BSD	interface	Permissive
litesata	librecores	LiteSATA provides a small footprint and configurable SATA gen1/2/3 core whose aim is to lower entry level of complex FPGA cores used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller	2-clause BSD	interface	Permissive
litedram	librecores	LiteDRAM provides a small footprint and configurable DRAM core whose aim is to lower entry level of complex FPGA cores used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller	2-clause BSD	interface	Permissive
litepcie	librecores	LitePCIe provides a small footprint and configurable PCIe gen1/2 core whose aim is to lower entry level of complex FPGA cores by providing used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller	2-clause BSD	interface	Permissive
litejesd204b	librecores	LiteJESD204B provides a small footprint and configurable JESD204B core whose aim is to lower entry level of complex FPGA cores by providing used in today's SoC such as Ethernet, SATA, PCIe, SDRAM Controller	2-clause BSD	interface	Permissive
EurySpace	librecores	Space Communication System based on CCSDS recommendations	MIT	interface	Permissive

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
HDMI2USB	librecores	The HDMI2USB project develops affordable hardware options to record and stream HD videos (from HDMI & DisplayPort sources) for conferences, meetings and user groups.	2-clause BSD	interface	Permissive
USB 1.1 Device IP Core	librecores	USB 1.1 slave/device IP core derived from USB 2.0 Function IP core save that all the high speed support logic has been ripped out and the interface changed from shared memory to FIFO based	3-clause BSD	interface	Permissive
USB 2.0 Device IP Core	librecores	This is a USB 2.0 compliant core. Due to the high interface speed, an external PHY will be required and an industry standard PHY interface for USB has been developed. This interface is called USB Transceiver Macrocell Interface (UTMI) and is WISHBONE SoC compliant.	3-clause BSD	interface	Permissive
AES (Rijndael) IP Core	librecores	AES (Rijndael) IP Core (128 bit version)	3-clause BSD	interface	Permissive
NoC Implementation Written in SystemVerilog	librecores	This is a Network on Chip (NoC) Router/Fabric implementation written in SystemVerilog.	Apache v2.0	interface	Permissive
MIPI CSI-2 Receiver	librecores	This project is an open source (MIT license) MIPI CSI-2 receive core for Xilinx FPGAs, supporting 4k resolution at greater than 30fps.	MIT	interface	Permissive
Wishbone	librecores	Wishbone is an interconnect for Systems-on-Chip.	other	interface	
scct	librecores	SCCT is a Simple Capture/ Compare Timer written in Verilog. It provides multiple capture/compare channels that use a common counter.	GPL	component	Copyleft

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
libstorage	librecores	Library of RTL components for data storage	ISC	component	Permissive
The PicoBlaze-Library	librecores	The PicoBlaze-Library offers several PicoBlaze devices and code routines to extend a common PicoBlaze environment to a little System on a Chip (SoC or SoFPGA).	Apache v2.0	component	Permissive
PicoBlaze-Examples	librecores	PoC - “Pile of Cores” provides implementations for often required hardware functions such as FIFOs, RAM wrapper, and ALUs.	Apache v2.0	component	Permissive
The PoC-Library	librecores	PoC - “Pile of Cores” provides implementations for often required hardware functions such as Arithmetic Units, Caches, Clock-Domain-Crossing Circuits, FIFOs, RAM wrappers, and I/O Controllers.	Apache v2.0	component	Permissive
litescope	librecores	LiteScope is a small footprint and configurable embedded logic analyzer for use in an FPGA and aims to provide a free, portable and flexible alternative to large vendor solutions	2-clause BSD	component	Permissive
WISHBONE Interconnect IP Core	librecores	This is a WISHBONE Interconnect Matrix IP core.It can interconnect up to 8 Masters and 16 Slaves.	3-clause BSD	component	Permissive
sha256	librecores	Hardware implementation of the SHA-256 cryptographic hash function with support for both SHA-256 and SHA-224	2-clause BSD	component	Permissive

Name of Project	Where Project is Recorded	Brief Description	Type of Licence	Category	Licence Type
siphash	librecores	This is a hardware implementation of the SipHash [1] keyed hash function written in Verilog 2001 and is designed as a self contained core that performs the message block processing including initialization, compression and finalization operations.	2-clause BSD	component	Permissive

Table 7: LibreCores

Appendix 5

Analysis

		OpenCores	Librecores	Total
Type of project	Processor Core	9	14	**23**
	Component	3	9	**12**
	Interface	11	14	**25**
TOTAL:		**23**	**37**	**60**

Table 8: Summary Analysis

		OpenCores	Librecores	Total
Licences	Copyleft	19	5	**24**
	Permissive	3	30	**33**
	Other	1	2	**3**
Total:		**23**	**37**	**60**

Table 9: Licence Analysis

		Processor Core	Component	Interface	Total
OpenCores	Copyleft	7	3	9	**19**
	Permissive	1	0	2	**3**
	Other	1	0	0	**1**
Total:		**9**	**3**	**11**	**23**

Table 10: OpenCore Analysis

		Processor Core	Component	Interface	Total
Librecores	Copyleft	4	1	0	**5**
	Permissive	9	8	13	**30**
	Other	1	0	1	**2**
Total:		**14**	**9**	**14**	**37**

Table 11: Librecore Analysis

		Processor Core	Component	Interface	Total
	Copyleft	11	4	9	**24**

		Processor Core	Component	Interface	Total
Both	Permissive	10	8	15	**33**
	Other	2	0	1	**3**
Total:		**23**	**12**	**25**	**60**

Table 12: Analysis of Opencores and Librecores