Open Source Compliance for Embedded Systems: a Closer Look at the GPL-2.0 and its Requirements

Dr. Hendrik Schöttle [a]

(a) Rechtsanwalt and Partner at Osborne Clarke, Munich, Germany

DOI: 10.5033/jolts.v11i1.136

Abstract
Open source compliance has achieved the next level: In the beginning of open source compliance actions, violations of open source licenses have been enforced by the community as a matter of principle. At the same time, awareness about compliance requirements have also risen, especially when distributing open source software as part of embedded systems. But is compliance nowadays more difficult than ever? And how does this apply especially for embedded systems?
Thoughts of a German qualified lawyer who has been involved in several compliance cases with a focus on the developments concerning GPL-2.0 compliance in Germany.

Keywords
Law; information technology; Free and Open Source Software; Open Source Compliance; GPL-2.0; GPL; Embedded Systems

1. Open Source Licenses as Playground for Trolls

Anyone who distributes open source software (OSS) as part of a product has to comply with its license conditions. Even though many companies avoid OSS, and even though in many cases I still see it wrongly assumed that no OSS is used: Today, most players dealing with OSS have realized that there are licensing conditions which entail special obligations and that these obligations generally have to be fulfilled.

In many cases, however, I still see helplessness with regard to which obligations have to be fulfilled in detail and what this means from a practical point of view.

This would not be an issue, as in most cases OSS communities and representatives of proprietary software get along better than just a few years ago. Many companies have realized that participating in OSS projects leads to a competitive advantage. OSS projects are no longer forked by companies, i.e. they are no longer developed independently from the original OSS project. Many companies give improvements, patches and bug fixes back to the community. Business models around OSS have also changed: A rising number of projects does not focus on the principle of free software anymore; many pursue quite openly economic interests - however, not with the sale of software, but with the services behind it.

Nevertheless, at least in Germany the number of cases, in which missing OSS compliance is enforced, is increasing. Initially, such enforcement of OSS license violations was driven by idealistic motives. Nowadays, it is often about financial interests of individuals. The majority of cases do not end up in court, but are settled outside of court.

Although many companies do know that OSS compliance is important, organizations and individuals when using OSS have problems to understand and to implement license compliance. Why is that? The main problem is likely to be the OSS licenses themselves. Many of them were drafted in the 1990's, some even in the 1980's. Even if the main principles of programming have remained the same, software development and its distribution have changed considerably over the years.

As a result, today we are dealing with OSS licenses whose obligations require expert knowledge and support in order to be fulfilled in practice with reasonable efforts. Ultimately, any company that distributes OSS components is thus exposed to potential injunctive relief from the copyright holders. This is repeatedly countered by OSS communities, who argue that in practice there is no need to fear legal disputes, as everyone gets along well with each other. However, this is only small comfort for those over whom the sword of Damocles is hanging. From a legal risk point of view, a single developer of the referring OSS is able to use injunctive relief against companies, prohibiting further distribution of their products with immediate effect, simply for violating formal OSS license obligations. The other developers' assurance that they will not take any legal action is irrelevant.

Let me get to the point: Many open source licenses used in practice have a potential for "trolling". They require compliance with conditions that can simply no longer be met in times of nowadays' software development with reasonable efforts - and thus offer the same opportunities that "patent trolls" do have. At least in Germany, the OSS community is now threatened with the same phenomenon that it has been fighting for years: the prosecution of violations of the law for commercial interests. This time, however, not by means of copyright and proprietary license agreements, but by means of OSS licenses.

In the following, the example of software which is licensed under the GNU General Public License, Version 2.0 (GPL-2.0), and used in embedded systems will show why the implementation of information obligations can be difficult in single cases. An Internet router may serve as an example. Its operating system is based on the Linux kernel, but unlike a classic desktop computer, the user has only limited access to the operating system. Of course, there is not only the GPL-2.0 as open source license – many more different licenses do exist. However, as the GPL-2.0 is one of the licenses with strict obligations and as it is enforced in practice repeatedly, it may serve as a good example.

2. The Story so Far

2.1. OSS Compliance 1.0

A little more than ten years ago, legal experts in Germany still disagreed as to whether OSS licenses were legally enforceable at all. Licenses such as the GPL provide for a transfer of rights which is conditioned by the fulfilment of several obligations - a legal construct which at first sight runs against many legal systems such as those established by the German Copyright Act.

In 2004, the GPL-2.0 was tested in court for the first time. The Munich Regional Court did not only decide that the construct of the conditional transfer of rights is effective and does not constitute an unreasonable disadvantage in the sense of the German law on standard terms and conditions (which would have led to its unenforceability).[1] The court also held that a missing license text and missing source code is a breach of the GPL-2.0's obligations, which can lead to injunctive relief claims.

In the period that followed, missing license texts and missing source codes of GPL-licensed components were repeatedly enforced before court. Many of these enforcements were driven by idealistic motives: In the course of increasing software modularization, more and more OSS components were used in commercial products, but not declared as such. After an initial phase, where OSS was not taken seriously, it soon conquered the world and found its way into commercial products. The community wanted to be taken seriously and this included fulfilling the OSS licenses' requirements.

2.2. OSS Compliance 2.0

In the initial phase, the dispute over compliance with OSS licenses was mainly about missing license texts and source codes, but soon it was about more: In 2012, the Bochum Regional Court ruled in a landmark decision that even if OSS was basically free of charge, damage claims were possible according to the principles of license analogy.[2] This answered the question of whether it was just a matter of principle or money.

Moreover, compliance was about more obligations than providing the license texts. Suddenly it was also about missing copyright notices, about a missing or insufficient offer of the source code or about a missing disclaimer - obligations, whose compliance even large parts of the OSS community had not taken seriously for quite a while.

Not surprisingly, there are now disputes within the OSS community about how to properly deal with license violations. Only recently, a developer was expelled from a development team; he was blamed of pursuing his own financial interests.[3]

Additionally, disputes over OSS license infringements are no longer limited to the classic relationship in which a right holder of OSS licenses takes action against an infringer. Nowadays, there are also disputes between companies about the effect of OSS components used, in the development of which the companies were not directly involved. For example, in the *Versata vs. Ameriprise* case before a US court, a software vendor and its customer disputed whether the copyleft effect of a GPL-licensed OSS component results in the proprietary software distributed by the software vendor being able to be used as OSS free of charge.[4] This further increases the risk of being taken to court for license violations.

3. The Obligations of the GPL in Detail

3.1. Delivery of the License Text

First of all, the GPL requires that a copy of the license text be provided with the licensed software. Specifically, the GPL-2.0 requires the following in paragraph 1:

[1] Regional Court of Muenchen, Judgement of May 19th 2004 – File number 21 O 6123/04.

[2] Regional Court of Bochum, Judgement of March 3rd 2016 – File number I-8 O 294/15. In case of a license analogy, damage claims are calculated on the basis of a fictitious or existing license fee that could be charged for legally compliant licensing, usually added by an additional surcharge.

[3] *H. Meeker*, Patrick McHardy and copyright profiteering, accessed at https://opensource.com/article/17/8/patrick-mchardy-and-copyright-profiteering

[4] United States District Court of Travis County Texas, Judgement of August 13th 2008 Case No. A-14-CA-12-SS; Versata vs. Ameriprise.

> *"You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you […] give any other recipients of the Program a copy of this License along with the Program."*

The GPL-2.0 dates back to 1991 - so let's go back to a time when software was distributed in cardboard boxes on 5.25" floppy disks together with a printed user manual. Software did not consist of too many sub-components, which made it feasible to list all of them in the user manual together with a handful of copyright notices. In this case, a copy of the license text could easily have been included in the manual to fulfill the obligations.

Today, the situation has changed. Software packages have grown from a few megabytes to gigabytes in many cases. Even smaller software contains hundreds of sub-components while in the past in many cases storage space was the limiting factor. Additionally, software is in most cases distributed without a physical copy. Even though boxed software still exists, it has been outpaced by downloaded software by far. As a consequence, the number of different license texts applicable to software included in a product has grown as well.

What about embedded systems? Let's think about the Internet router mentioned above. Instead of a manual, there might be a leaflet in the package that contains a short manual of a few lines of text and refers to a website for further information. How can the obligation to pass on the license text be fulfilled?

A reference to a website is not sufficient - the wording "give [...] a copy of this license along with the program" is clear in this respect; by the way, this has also been confirmed by the Munich Regional Court.[5]

It may be possible to include the license text in paper form in case of one single license. However, today many products contain numerous third-party components under dozens of different licenses, each of which requires the distribution of the license text. This quickly leads to a document that can contain several hundred pages of text consisting of the numerous different license texts and copyright notices (see below 3.3) contained in the product. Especially in the case of low-priced products produced at high quantities, the delivery of an additional paper copy can make the distribution of the product considerably more expensive - or make it economically impossible in comparison to competing products.

An alternative may be to show the license text on the display of the device - if the device has one. However, it becomes difficult if such a display is missing. The Free Software Foundation, which was involved in the drafting of the GPL, is of the opinion that it is sufficient to reproduce the license text via the interface used by the device. In its FAQ, the Free Software Foundation wrote the following:

> *"Q: My program has interactive user interfaces that are non-visual in nature. How can I comply with the Appropriate Legal Notices requirement in GPLv3?*
>
> *A: All you need to do is ensure that the Appropriate Legal Notices are readily available to the user in your interface. For example, if you have written an audio interface, you could include a command that reads the notices aloud.*[6]

In the opinion of the Free Software Foundation, it would therefore be sufficient if, for example, a headset reads the license text aloud. Even though the Free Software Foundation is definitely an institution that cannot be underestimated in questions relating to the GPL, one may also have a different

[5] Regional Court of Muenchen I, Judgement of May 12th 2007 – File number 7 O 5245/07.

[6] Frequently Asked Questions about the GNU Licenses – https://www.gnu.org/licenses/gpl-faq.html#NonvisualLegalNotices. This FAQ refers to the GPL-3.0. However, as Clause 4 GPL-3.0 is identical with Clause 1 GPL-2.0, we see strong arguments that the FSF's statement can be applied accordingly to the GPL-2.0.

view on this. It can be doubted whether a voice recording can be regarded as a provision of a license text, leaving alone the comprehensibility of the acoustic transmission of such texts.

If however, following the opinion of the Free Software Foundation, this is sufficient, then other interfaces could very likely also be used. For example, a device is connected with a local network by the user by connecting with an external device such as a smartphone or notebook computer to a Wi-Fi hotspot set up by the device. In a second step, the user then enters the necessary data on a web server run on the device (as this is the case with almost all Internet routers). In this case, it should be sufficient to store the license text locally on this web server within the embedded system. This would not qualify as a link to an external source in this case, the license text would rather be stored on the same storage medium as the software.

Another solution may be the provision of a CD-ROM or a DVD containing the necessary information. However, in this case, it has to be considered whether the recipient will be able to read data from such CD-ROM or DVD. This is another good example for changed requirements due to technical developments: While 15 years ago, in private households, an Internet router will very likely have been the gateway between the Internet and a desktop computer, in most cases equipped with a CD drive, nowadays many households are only using smartphones or tablet computers only and do not have the possibility to read CD-ROMs or DVDs any more. As a result, such provision may be sufficient in a B2B scenario where such equipment is more likely to be present at the recipient or in a B2C scenario in cases where the existence of a respective drive is likely.

In theory, the above solutions sound quite simple. In practice, however, considerable efforts must be invested to find the required license information. Software components that are integrated into the own software often contain further subcomponents, which again contain subcomponents, and so on. The Linux kernel alone brings up more than 60,000 files, in which hundreds of license texts are hidden, which are available in various formats. Furthermore, files can have multiple licensing statements. There are scanning tools which are used by many organizations in the context of compliance processes and which search the source code for license texts or single copyright notices. However, persons must be aware that in some cases, experts need to correct the results generated by these tools. Standardization of how to express license and copyright information is still emerging; as such, many OSS today did not implement it so far[7]. Anyone who wants to ensure that any and all license texts have been captured correctly will have to depend on manually reworking the scan result.

Additionally, as even complex software has sneaked into everyday products, license requirements may have to be fulfilled where a distributor does not even think of software being contained in a product.

Thus, license compliance is not that easy, as software is nowadays distributed in many more ways than about 30 years ago and as software contains much more sub-components than at that time. As the hardware has become cheaper as well, it can become an economical hurdle to provide written documentation together with a low-priced product.

3.2. Reproduction of Copyright Notices

In addition to the delivery of the license text, GPL-2.0 also requires the reproduction of copyright notices. Clause 1 of the GPL-2.0 states with this regard:

> *“You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; […].”*

[7] See e.g. the FSFE‘s reuse.software project at https://reuse.software

Even though the handling of copyright notices under GPL-2.0 seems like a classical obligation, a closer look at the license texts shows that its requirements are far from being clear:

It is already unclear, whether the distribution of software requires to attach a copyright notice, even when distributing unmodified software or whether it is sufficient to retain existing copyright notices. The language of Clause 1 states that one has to "*publish on each copy an appropriate copyright*". Taking this requirement literally, one would have to add a copyright notice in any case, even when not modifying the software. Such interpretation could be backed with regard to the following clause of the GPL-2.0, which requires to "*keep intact all notices that refer to this License*": Obviously, the GPL-2.0 differentiates between the "*publication*" of a notice and "*keeping a notice intact*". However, this can hardly be meant as a requirement when distributing unmodified software. In such a case, the distributor cannot be regarded as copyright holder of the software, as no changes would be made to the existing code. So in this case, it would not only be unnecessary to add one's own copyright notice, but it would also be misleading. However, one could still interpret this clause requiring to add a copyright notice of upstream copyright holders. In this case, a distributor would have to do research on possible upstream copyright holders and complete missing information. However, even though such interpretation could be backed by the language of Clause 1 as well, it would be error prone. Thus, in German literature, experts advise not to add any copyright notices to unchanged code, they hold that it is sufficient to reproduce existing copyright notices.[8] This seems to be the most convincing result and also what is seen as common practice, even though it may be difficult to overcome the license's language which is stricter, as it clearly differentiates between "*publishing*" and "*keeping intact*" notices.

The vast majority of copyright notices can be found in the individual source files. The single programmers include a note concerning their contributions there in form of a comment. However, these comments are removed when compiling the software into binary code. The binary code usually only contains a general copyright notice in the software, which is displayed when the software is executed. All notices contained in the source code are missing.

Firstly, it is almost impossible to compile the copyright notices when passing on OSS in binary form only. Why is this so? So is it necessary to extract such comments from the source code and to compile them in a separate document in order to ensure their delivery with the binary code? Let us take a closer look at the license: The Problem is the reference chain from Clause 3 to Clause 1. Clause 1 deals with the distribution of the source code. When it comes to distribution of the binary code as described in Clause 3, this Clause simply refers to Clause 1:

> *"You may […] distribute the Program […] in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following […]"*

Following this referral to Section 1, when distributing the software in object code, one again would have to "publish" a copyright notice. As outlined above, it is probably sufficient to retain existing copyright notices, even though the language of Clause 1 suggests more. However, as during the compilation process, the copyright notices are removed from the code, one may very well argue that they have to be reinstated, especially as Clause 1 speaks of "publishing" such notices, not of "keeping them intact". According to German legal experts, the requirement to extract copyright notices from the source files can be backed by a strict interpretation.[9] Other voices point out that increasingly more copyright holders insist of exactly doing that.[10]

[8] O'Reilly, Die GPL kommentiert und erklärt, marginal 32.

[9] *Jaeger/Metzger*, Open Source Software, 4th Ed. 2016, marginal 37.

[10] *Hemel*, Practical GPL Compliance, p. 39.

Thus, there is a risk that a copyright holder may require said extraction of copyright notices from the source files when distributing the binary, even though such extraction does not seem to be common practice.

So if you follow the strict approach, it would be necessary to collect the copyright notes from the source files and merge them into a separate document. If you have several software components, you also have to assign the notes to individual files additionally. With regard to the Linux kernel already mentioned, this can sum up easily to several thousands of source files, which then have to be searched individually for copyright notes. By the way, this obligation applies regardless of any modifications to the software. Thus, anyone who distributes an unmodified Linux distribution in binary code must go through this exercise, unless the necessary information has already been provided. Even though some scanning tools are specialized in collecting such copyright notices, naturally they do not find everything and require manual corrections or reviews.

Additionally, the formal requirements for copyright notices are stricter than for the delivery of the license text. The Free Software Foundation refers to the fact that it is generally required to deliver the source code anyway and that a delivery without source code is only permitted in limited exceptions.[11] Since the copyright notices are contained in the source code, isn't it enough to simply deliver the source code with the product in order to fulfill the requirements? Well, paragraph 1 of the GPL-2.0 requires that “an appropriate copyright notice” be published, namely “conspicuously and appropriately”. So a conspicuous and appropriate publication of a copyright notice is required, which again has to be appropriate.

Let's go back in time to the early nineties: Software is distributed on floppy disks or floppy disk images shared via the Usenet. The software user inevitably has access to a floppy disk drive and a desktop computer - because the software cannot be used in any other way. So if the source code is also delivered with the binary code, the user can read it with a simple text editor and has the possibility to view all copyright clauses. Accordingly, legal experts hold it as sufficient to deliver the copyright notices together with the software.[12]

So can the source code of embedded systems simply be stored together with the binary files in the memory of the device in order to fulfill the information obligations? This case is somewhat different from the desktop computer: With most embedded devices, the user does not have full access to the file system. The user cannot access and read source files stored on the embedded device when using only the means of the embedded device. The information may be stored on the device. Whether it is “appropriate” may be doubted, but they can probably not be considered as catching the eye in the sense of the term “conspicuously”. As outlined above in section 3.1, it depends on the individual case whether it is sufficient to include a data carrier such as a DVD or a flash drive. It may be compliant if it is sufficiently certain that the typical user of the embedded device has an appropriate reader, i.e. a desktop computer or a notebook with USB ports. In many cases, however, one may not this will no longer be mandatory.

The Free Software Foundation however even goes one step further. In response to this article’s author's question, whether the copyright clauses had to be extracted manually from the source files if only the binary files were distributed, they replied:

> *“We would describe a written offer as distributing the source […] Delivery of binaries with just a written offer […] is perfectly acceptable. […] I’m also not aware of someone putting in place the practice you describe.”* [13]

[11] https://www.gnu.org/licenses/gpl-faq.html#UnchangedJustBinary .

[12] *Koglin*, ifrOSS, Die GPL kommentiert und erklärt, 2005, p. 50 marginal 36.

[13] Quote from an e-mail of the Free Software Foundation dated August 22nd 2016.

Thus, according to the Free Software Foundation, it should even suffice if the copyright notices are not provided at all, but if only an offer is made to receive the source code. This would then be equivalent to the delivery of the source code and thus the information obligations would also be fulfilled.

The GPL does indeed allow a written offer addressed to the general public to receive the source code instead of its distribution. However, this only applies to the source code and not to the copyright notices. The Free Software Foundation may take a liberal view here - but the wording of the GPL-2.0 alone does not reflect this view. Even if there are good arguments for such an interpretation of the GPL-2.0, an uncertainty remains.

Does the wording of the GPL-2.0 still matter now, where the Free Software Foundation obviously takes a different view? In the OSS community, the Free Software Foundation is often compared with a legislator, since it contributed significantly to the wording of the GPL and thus also claims the sovereignty to interpret it. The view of the Free Software Foundation may be an important one, but it is not necessarily the authoritative one. The GPL is not a law, but a template. And if this template is used between two contracting parties, at least under German law, any ambiguities concerning its interpretation are resolved by interpreting firstly the wording of the text, secondly by the underlying understanding of the parties and only in last instance by referring to possible legal opinions of third parties.

There are good arguments that one may interpret the GPL's requirements in accordance with the Free Software Foundation restrictively and come to the result that, despite the opposite wording, due to the changed form of the software distribution a restriction of the information duties is required, so that with the distribution of the binary files no copyright notices must be compiled manually from source files and be delivered with the binaries. Even if there are no references to such an interpretation in literature and case law, one still has the Free Software Foundation on its side.

3.3. Reproduction of a warranty disclaimer

Besides the copyright notices, Clause 1 of the GPL-2.0 also requires the provision of a warranty disclaimer. And again, even though this obligation as well seems to be pretty standard, like the copyright notices, it shows an inconsistency that leads to uncertainty and possible compliance issues in practice.

As outlined above, Clause 1 of the GPL-2.0 requires to publish on each copy a warranty disclaimer. This obligation makes sense in case of source code provision – and it becomes clear when taking a look at the Section "How to Apply These Terms to Your New Programs", which is not part of the GPL-2.0's license text itself. With regard to the notices, including the warranty disclaimer, this section recommends "*to attach them to the start of each source file to most effectively convey the exclusion of warranty*". Such procedure makes sense in order to ensure an increased level of comfort regarding the warranty exclusion – even though strictly speaking such additional disclaimer would not be necessary, as the GPL-2.0 as such already contains a warranty disclaimer in Clauses 11 and 12.

However, due to the reference chain from Clause 3 to Clause 1 in case of binary distribution, the obligation concerning the warranty disclaimer does not make sense any more: There is no source code, in whose header section a warranty disclaimer could be integrated. However, the GPL-2.0 still requires its provision in addition to the warranty disclaimer which is already contained in the license. As a result, a distributor of binary code has to provide the license text of the GPL-2.0 and an additional warranty disclaimer. As such disclaimer, one may use a verbatim copy of the boilerplate example of a warranty disclaimer which can be found at the end of GPL-2.0's license text. Even more confusing: the provision of the complete license text together with this boilerplate disclaimer would not be sufficient – as the warranty disclaimer is only an example, but not a disclaimer.[14]

[14] The GPL-2.0 only suggests to "attach the following notices to the program", but it does not attach this sample text as actual disclaimer.

As a result, when distributing GPL-2.0'ed binary code, one has to add an additional warranty disclaimer at the end of the license text or elsewhere, in order to comply with formal requirements of GPL-2.0, even though it is hard to see an added value in providing this additional disclaimer.

This is by the way an issue that is not properly handled in many open source projects as well. Even the Linux kernel does not contain an additional warranty disclaimer. Such disclaimers can be found in single source files of the Linux kernel here and there – but not in all of them. As there is no general *additional* warranty disclaimer on a higher level in the Linux kernel package,[15] even the distribution of the unmodified kernel source files would constitute an infringement of the GPL-2.0, as for those source files without an additional warranty disclaimer the requirements of the GPL-2.0 are not fulfilled. As outlined above, the disclaimer included in the license text as such is not sufficient.

I would be glad if I could state that this is only a theoretical issue when following a very strict interpretation of the GPL-2.0. However, I have advised cases in practice where distributors have been subject to copyright infringement claims based exactly this issue: a missing additional warranty disclaimer in case of distribution of GPL-2.0'ed software, even though the license text including a warranty disclaimer had been provided together with the software.

There are good arguments that an additional warranty disclaimer is not necessary with regard to the purpose of the GPL-2.0, as the disclaimer already included in the license text is sufficient – it has to be provided as part of the license text anyway. However, strictly following the literal interpretation of the license text, an additional warranty disclaimer cannot be avoided.

3.4. Availability of the source code

Finally, section 3 of the GPL-2.0 requires the availability of the source code. This obligation can either be fulfilled by delivering the source code directly with the product[16] or by making a written offer to give any third party the source code on a medium customarily used for software interchange in return for reimbursement of the costs of reproduction, such offer to be valid for a period of at least three years.[17] In case of a non-commercial distribution one also has the possibility to refer to a respective offer one has received,[18] however, this possibility is excluded in case of a commercial distribution.

In practice, this means either the delivery of the source code or the submission of a written offer. At least under the widespread GPL-2.0 it is not sufficient to refer to a download option, as rather the source code must be provided on a standard data carrier. Of course, it does not hurt to refer to a download option in addition, as this will be the standard way in practice. However, from a legal point of view, if one does not provide the source code directly with the binary, one cannot avoid such an offer. It should be noted that such an offer cannot be found in the text of the GP-2.0 itself – thus, it has to be added to the documentation when distributing the software.

The Free Software Foundation provides a template text of such an offer, which is limited to three years and in its opinion meets the license requirements.[19] However, when adopting this text, it should be checked whether the time limit of three years for the offer can be accepted in the specific case. Some other open source licenses contain similar obligations to make such an offer, but for an indefinite period.[20] Anyone who now accepts the time limit proposed by the Free Software Foundation

[15] When taking a look at the latest stable release 5.5.1, downloaded from kernel.org at the time of publication of this article, the file *COPYING* only referred to the files *LICENSES/preferred/GPL-2.0*, *LICENSES/exceptions/Linux-syscall-note* and *Documentation/process/license-rules.rst*, none of them containing an additional warranty disclaimer.

[16] Sec. 3 a) GPL-2.0.

[17] Sec. 3 b) GPL-2.0.

[18] Sec. 3 c) GPL-2.0.

[19] https://www.softwarefreedom.org/resources/2014/SFLC-Guide_to_GPL_Compliance_2d_ed.html#appendix-1-offer-of-source-code

[20] For example, see Sec. 3.1 of the CDDL-1.0/CDDL-1.1.

must ensure that it only applies to the GPL-licensed components and not to other licenses, as otherwise a violation of the obligation to submit an unlimited offer is given. In addition, one risks of violating the wording of the GPL-2.0 even if using the sample text if it is not clear when the three-year period begins.

If the source files are delivered on the embedded device - which should be fine at first sight - it must nevertheless be ensured that copyright notices and any warranty disclaimers contained in the source code, which must be made available in a conspicuous and appropriate manner, can also be perceived in such a conspicuous way, see 3.3 above.

4. Conclusion

As a result, when following a strict interpretation, the obligations of GPL-2.0 lead to relevant efforts for ensuring license compliance when interpreting it strictly – not only in case of embedded systems:

For example, anyone who distributes a Linux distribution in binary code without modifications, usually commits a license violation if the source code is missing. Following a strict interpretation of the GPL-2.0, the distribution of the binary code requires extraction of the copyright notices from the source code.

But even in case of distribution of the unmodified sources, in many cases a license violation would be given, as an additional warranty disclaimer is missing, as e.g. in case of the Linux kernel.

Even those who deliver the source code with an embedded system are at risk of not fulfilling some obligations. The copyright notices must be appropriate and conspicuous. Even if there are good arguments for a restrictive interpretation of the GPL-2.0: one may doubt that this obligation is fulfilled if the source code cannot be accessed by the user of the embedded system without additional means.

To boil it down, distribution of open source software, specifically of GPL-2.0 licensed software faces two important tasks, when interpreting the GPL-2.0 strictly: First of all, it is not possible to distribute software without adding additional information, be it missing copyright notices in case of binary distribution or a missing warranty disclaimer in most cases of software distribution, regardless of whether in source or in binary form.

Second, the GPL-2.0 is focused on source code distribution, which may have been the best choice in early days' distribution of desktop software and is still important to the open source community - but is something that the majority of end users of embedded systems does not care about. The typical user of an embedded system does not want to modify the firmware or compile it in order to get the embedded system running.

Of course, one may say that it is simply the main intention of the GPL-2.0 to distribute respectively licensed software in source code form and in binary form as an exception. Nonetheless, it would be helpful to set up clear guidelines for binary code distribution and to include all necessary license compliance information into the OSS distribution once from the start instead of requiring every single distributor of such software to complete and edit missing information.

For some few developers who have contributed to OSS, OSS compliance has proven to be a goldmine - to the chagrin of the community, which simply wants to develop good software, and to the chagrin of the industry, which faces requirements that are not easily met without dedicated effort and that are disproportionate to the benefits.

Many companies are burying their heads in the sand in view of the efforts entailed by compliance requirements and hope that they will not get hit.

From the perspective of the advising lawyer one may cynically conclude that everything is fine, the need for advice is sustainably secured for the near future. But it would hardly be in the interest of the involved stakeholders if the distribution of GPL-licensed software was to reach legal limits due to technical developments. The involved stakeholders should stick their heads together and find a solution that gives both sides legal security. The good thing is that this already happens with regard to certain projects.[21] Such efforts to clean up OSS projects should be intensified and cover more and more projects. After all, open source software should not serve individuals, but the community.

So what to do in practice in the meantime? Best practice in case of binary distribution of GPL-2.0 licensed software in embedded systems is probably the extraction of copyright notices from the sources, addition of a warranty disclaimer and provision of the written offer for the source code. In case the extraction of copyright notices is considered too burdensome, one may refer to the FSF's statement on the written offer of the sources containing the copyright being sufficient with this regard as well. Concerning the form of provision, the best and in most cases most expensive way would be a printed copy, delivered together with the product. A second-best solution would be accessibility on the product's electronic display, if any. A fallback solution may be the provision in form of documentation locally stored in an internal web server, accessible via a Wi-Fi hotspot. A mere reference to an external online resource is not sufficient, at least according to the license text of the GPL-2.0, which has been confirmed by German case-law. In any case, a software distributor will have to ensure by technical and organizational measures that the OSS licenses' requirements are complied with – be it manually or by relying on automated OSS toolchain solutions.

The world has changed since the 1990's and so have the technical requirements for software distribution. The distribution of software as part of today's embedded systems shows that problems and questions have emerged which have to be addressed. Hopefully, this article helps to identify some of them and to provide at least workarounds, if not fixes for them.About the authors

About the Author

Dr. Hendrik Schöttle*, Rechtsanwalt, Partner, Specialized Lawyer in IT law, is Partner in the Munich Office of Osborne Clarke in Germany. He advises on IT law and data protection law, with a focus on software licensing models, in particular relating to open source software. He is frequently named in lawyer rankings in the field of IT law; the German JUVE handbook 2019/2020 recommends him as "leading name in the area of open source". Hendrik became a lawyer in 2005 and joined the Munich office of Osborne Clarke in 2007. He spent a number of years as a software developer at the Institute for Legal Informatics of Saarland University. He is the author of numerous publications and co-author of several handbooks and commentaries on IT law related topics, a lecturer in IT law at the Deutsche Anwaltakademie (German Lawyers' Academy) and a frequent speaker on topics relating to IT law. Hendrik is a board member of the BITKOM working group Open Source, a member of the Data Protection Law Committee of the German Federal Bar Association, a member of the information technology working group of the German Lawyers Association (DAV), and a member of the German Society for Law and Computer Science. He can be reached at hendrik.schoettle@osborneclarke.com*

[21] *Corbet*, SPDX identifiers in the kernel, accessed at https://lwn.net/Articles/739183/ for further reference.

Licence and Attribution

This paper was published in the Journal of Open Law, Technology, & Society, Volume 11, Issue 1 (2019). It originally appeared online at http://www.jolts.world

This article should be cited as follows:

Schoettle, Hendrik (2020) 'Open Source Compliance for Embedded Systems: a Closer Look at the GPL-2.0 and its Requirements', Journal of Open Law, Technology & Society, 11(1), pp 75 – 86
DOI: 10.5033/jolts.v11i1.136